



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**IMPLEMENTASI PEMBANGKITAN SECRET KEY
BERBASIS RECEIVED SIGNAL STRENGTH (RSS)
UNTUK SISTEM KOMUNIKASI BAWAH AIR**

Tyas Basuki Priambudi
NRP 07111440000063

Dosen Pembimbing
Dr. Ir. Wirawan, DEA.
Sri Rahayu, S.T., M.Kom.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[halaman ini sengaja dikosongkan]



FINAL PROJECT - EE 184801

**IMPLEMENTATION OF RECEIVED SIGNAL STRENGTH
(RSS) BASED SECRET KEY GENERATION FOR
UNDERWATER ACOUSTICS COMMUNICATION
SYSTEM**

Tyas Basuki Priambudi
NRP 07111440000063

Supervisors
Dr. Ir. Wirawan, DEA.
Sri Rahayu, S.T., M.Kom.

DEPARTEMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Implementasi Pembangkitan Secret Key berbasis Received Signal Strength (RSS) untuk Sistem Komunikasi Bawah Air” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2019



Tyas Basuki Priambudi
NRP 07111440000063

[halaman ini sengaja dikosongkan]

**IMPLEMENTASI PEMBANGKITAN SECRET KEY
BERBASIS RECEIVED SIGNAL STRENGTH (RSS)
UNTUK SISTEM KOMUNIKASI BAWAH AIR**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro
Pada**

**Bidang Studi Telekomunikasi Multimedia
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I,



Dr. Ir. Wirawan, DEA.
NIP. 196311091989031011

Dosen Pembimbing II,



Sri Rahayu, ST., M.Kom.
NIP. 196802281997022001



[halaman ini sengaja dikosongkan]

IMPLEMENTASI PEMBANGKITAN SECRET KEY BERBASIS RECEIVED SIGNAL STRENGTH (RSS) UNTUK SISTEM KOMUNIKASI BAWAH AIR

Tyas Basuki Priambudi
07111440000063

Supervisor I : Dr. Ir. Wirawan, DEA.
Supervisor II : Sri Rahayu, ST., M.Kom.

ABSTRAK

Underwater Acoustic Communication System (UACS) adalah teknologi komunikasi yang banyak digunakan di dunia kemaritiman, namun dewasa ini mulai dieksplorasi untuk kepentingan sistem komunikasi data dengan medium bawah air. Hal ini didorong kenyataan bahwa pemakaian gelombang radio (medium udara) di atas perairan memiliki atenuasi yang lebih besar oleh berbagai faktor peredam, sehingga medium air menjadi alternatif yang cukup potensial. Meskipun demikian, faktor keamanan (*security*) tetap menjadi pertimbangan penting. Ada banyak metode untuk membangun sistem keamanan komunikasi (termasuk di bawah air), diantaranya dengan teknik kriptografi seperti enkripsi (*ciphertext*), autentifikasi *user* dan/atau *content*, yang kesemuanya membutuhkan *secret key* yang kuat.

Pada tugas akhir ini, akan diimplementasikan algoritma berbasis *Received Signal Strength (RSS)* untuk pembangkitan *secret key* pada sistem *UACS* sederhana, dengan memilih nilai *RSS* tertentu untuk kepentingan sebagai link komunikasi hanya untuk dua pihak itu saja, yang tentu berbeda dari nilai *RSS* link lain. Pembahasan dalam tugas akhir ini meliputi perencanaan link propagasi bawah air, implementasi *UACS* sederhana di kolam Hidrodinamika ITS, serta pembuatan simulasi pembangkitan *secret key*-nya. Selanjutnya akan dilakukan analisa dan pembahasan unjuk kerja sistem *UACS* tersebut.

Dari hasil pengukuran dan pengolahan sinyal, dengan berbagai skema algoritma, dapat disimpulkan bahwa jenis sinyal Tone memiliki nilai Bit Mismatch Rate yang paling kecil, sedangkan jenis sinyal Noise memiliki level Bit Mismatch Rate yang paling besar.

Kata Kunci : UACS, Security, Autentifikasi, RSS, Secret Key.

[Halaman ini sengaja dikosongkan]

IMPLEMENTATION OF RECEIVED SIGNAL STRENGTH (RSS) BASED SECRET KEY GENERATION FOR UNDERWATER ACOUSTICS COMMUNICATION SYSTEM

Tyas Basuki Priambudi
07111440000063

Supervisor I : Dr. Ir. Wirawan, DEA.
Supervisor II : Sri Rahayu, ST., M.Kom.

ABSTRACT

Underwater Acoustics Communication System (UACS) is communication technology vastly used by the maritime domain until recently started to be explored for data communication purposes with the underwater medium. That is encouraged by usage of radio waves in water surface area is heavily attenuated by various factors, thus underwater medium made to be a potential alternative. However, the security factor must be taken as an essential consideration. There are many methods for building a communication security system (including underwater), such as encryption, user, and content authentication, which require a robust secret key.

In this final project, we will implement RSS based algorithm for the secret key generation in simple UACS, with choosing specific RSS value for communication between two parties only, which must be different with another link's RSS value. The study for this final project covers underwater propagation link planning, implementation of simple UACS in ITS Hydrodynamics pool, and simulation of secret key generation. Next analysis and study of UACS performance will be conducted.

From the results of signal measurement and processing, with various algorithm schemes, it can be concluded that the Tone signal type has the smallest Bit Mismatch Rate value, while the Noise signal type has the largest Bit Mismatch Rate level.

Keywords: UACS, Security, Authentication, RSS, Secret Key.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Dengan mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas limpahan rahmat dan berkat-Nya, sehingga penulis dapat menyelesaikan buku tugas akhir ini dengan judul:

“IMPLEMENTASI PEMBANGKITAN SECRET KEY BERBASIS RECEIVED SIGNAL STRENGTH (RSS) UNTUK SISTEM KOMUNIKASI BAWAH AIR”

Tugas akhir ini disusun sebagai salah satu persyaratan dalam menyelesaikan studi pada bidang studi Telekomunikasi Multimedia di jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam kesempatan ini, penulis ingin menyampaikan rasa terima kasih kepada pihak-pihak yang telah mendukung penulis selama proses menyelesaikan tugas akhir ini, khususnya kepada:

1. Kedua orang tua penulis yang sangat dicintai dan semoga dalam rahmat Allah SWT, Ibu Lusanti Karnasih dan Bapak Djatmiko Agus W yang telah mensupport banyak hal, mendoakan, dan merestui langkah anaknya.
2. Bapak Wirawan dan Ibu Sri Rahayu. Selaku dosen Pembimbing yang telah banyak memberi ilmu, pengarahan dan bimbingan selama penyelesaian tugas akhir ini.
3. Teman-teman Seperjuangan angkatan 2014, terima kasih atas semua kenangan dan kebersamaan selama masa perkuliahan di ITS.

Dalam penyusunan laporan tugas akhir ini penulis menyadari adanya keterbatasan. Oleh karena itu penulis sangat terbuka terhadap kritik dan saran untuk perbaikan karya tugas akhir ini.

Semoga buku Tugas Akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Departemen Teknik Elektro bidang Studi Telekomunikasi Multimedia pada khususnya. Dan lebih jauh diharapkan mampu memberi kontribusi terhadap perkembangan keilmuan, khususnya di bidang telekomunikasi.

Surabaya, Januari 2019

Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT.....	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metodologi	3
1.6 Sistematika	3
1.7 Relevansi / Manfaat.....	4
BAB II TEORI PENUNJANG.....	5
2.1 Sistem Komunikasi Akustik Bawah Air.....	5
2.2 Propagasi Gelombang Akustik Bawah Air.....	6
2.2.1 Pengaruh Persebaran Geometri.....	7
2.2.2 Dampak Penyerapan	8
2.2.3 Rugi-rugi Propagasi	9
2.2.4 Efek Gelembung Udara.....	9
2.3 Kriptografi.....	11
2.3.1 Enkripsi.....	12
2.3.2 Kunci	12
2.3.3 Pembangkitan Kunci	13
2.4 Perangkat Hydrophone	13

2.5 Speaker Bawah Air	14
2.6 Ilustrasi Sistem Komunikasi Bawah Air	15
BAB III METODOLOGI PENELITIAN	17
3.1 Diagram Alur	17
3.2 Perancangan Sinyal Akustik	18
3.3 Perancangan Algoritma Secret Key	19
3.3.1 Pendekatan Aono.....	19
3.3.2 Pendekatan Mathur.....	21
3.4 Rancangan Sistem Komunikasi Bawah Air	22
3.5 Perangkat Lunak Sistem Penerima.....	23
3.5.1 Fast Track Ultra.....	24
3.5.2 Adobe Audition CS6	25
3.5.3 Program Downsampling	25
3.6 Skenario Pengujian	27
3.6.1 Pengujian Algoritma.....	27
3.6.2 Pengukuran di lapangan.....	28
BAB IV ANALISA DATA DAN PEMBAHASAN	29
4.1 Tampilan Hasil Pengujian Sinyal dan Program	29
4.1.1 Hasil Pembangkitan Sinyal.....	29
4.1.2 Realisasi Pengujian Sinyal.....	31
4.1.3 Hasil Pengolahan RSS	31
4.1.4 Hasil Downsampling	32
4.2 Hasil Pengujian Algoritma	33
4.2.1 Hasil Pengujian Algoritma Aono.....	33
4.2.2 Hasil Pengujian Algoritma Mathur.....	34
4.3 Hasil Pengujian Lapangan	35
4.3.1 Hasil Pengolahan Algoritma Aono.....	36

4.3.2	Hasil Pengolahan Algoritma Mathur	36
4.4	Pembahasan	37
4.4.1	Rekapitulasi Skema Pengujian	37
4.4.2	Perhitungan Bit Mismatch Rate	38
4.4.2.1	Perbandingan BMR Algoritma Aono	39
4.4.2.2	Perbandingan BMR Algoritma Mathur	43
4.4.3	Perbandingan hasil pengukuran	46
BAB 5	PENUTUP	47
5.1	Kesimpulan	47
5.2	Saran	47
DAFTAR	PUSTAKA	49
LAMPIRAN A	51
LAMPIRAN B	53
LAMPIRAN C	63
BIODATA	PENULIS	87

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Sistem Komunikasi Bawah Air	5
Gambar 2.2 Skema Enkripsi Sederhana	12
Gambar 2.3 Contoh perangkat Hydrophone	14
Gambar 2.4 Speaker Bawah Air	15
Gambar 2.5 Blok Diagram Sederhana Sistem Komunikasi Bawah Air	12
Gambar 3.1 Flowchart Penelitian	17
Gambar 3.2 Interface Program Audacity	18
Gambar 3.3 Algoritma Aono	20
Gambar 3.4 Contoh pembangkitan bit dengan metode Aono	20
Gambar 3.5 Algoritma Mathur	21
Gambar 3.6 Contoh pembangkitan bit dengan metode Mathur	22
Gambar 3.7 Rancangan Sistem Komunikasi Bawah Air	22
Gambar 3.8 Blok Diagram Sistem Penerima	23
Gambar 3.9 Pengaturan pada Fast Track Ultra	24
Gambar 3.10 Pengaturan Adobe Audition CS6	25
Gambar 3.11 Kode Matlab untuk Algoritma Downsampling	26
Gambar 3.12 Blok diagram Downsampling	26
Gambar 3.13 Proses Pengujian Algoritma, (1) Recorder, (2) Downsampling, (3) Algoritma secret key, (4) Bit secret key terbangkitkan	27
Gambar 3.14 Skema Pengukuran Sistem Komunikasi Bawah Air	28
Gambar 4.1 Hasil Pembangkitan Sinyal	29
Gambar 4.2 Konfigurasi pengujian Sinyal	31
Gambar 4.3 Contoh Hasil RSS Sinyal (Tone)	32
Gambar 4.4 Contoh Hasil downsampling RSS Sinyal (Tone)	32

Gambar 4.5 Contoh Hasil ekstraksi bit dengan Algoritma Aono....	33
Gambar 4.6 (kiri) Transmisi A ke B dan (kanan) B ke A	34
Gambar 4.7 Hasil Pengujian Algoritma Mathur	35
Gambar 4.8 Hasil Ekstraksi Bit menggunakan Algoritma Aono	36
Gambar 4.9 Hasil Ekstraksi Bit menggunakan Algoritma Mathur	37
Gambar 4.10 Algoritma perbandingan Bit Secret Key	39
Gambar 4.11 Hasil Ekstraksi Bit algoritma Aono, sinyal Chirp, BMR = 0%	40
Gambar 4.12 Hasil Ekstraksi Bit algoritma Aono, sinyal Tone 5 kHz, BMR = 0%	40
Gambar 4.13 Hasil Ekstraksi Bit algoritma Aono, sinyal Tone 5 kHz & 8 kHz, BMR = 0%	41
Gambar 4.14 Hasil Ekstraksi Bit algoritma Aono, sinyal Tone 5 kHz bervariasi, BMR = 14,0625%	41
Gambar 4.15 Hasil Ekstraksi Bit algoritma Aono, sinyal DTMF, BMR = 26,5625%	42
Gambar 4.16 Hasil Ekstraksi Bit algoritma Aono, sinyal White Noise, BMR = 34,375%	42
Gambar 4.17 Hasil Ekstraksi Bit algoritma Mathur, sinyal Chirp, bit = 466; BMR = 7,94%	43
Gambar 4.18 Hasil Ekstraksi Bit algoritma Mathur, sinyal Tone 5 kHz, bit = 323; BMR = 0,929%	43
Gambar 4.19 Hasil Ekstraksi Bit algoritma Mathur, sinyal Tone 5 kHz & 8 kHz, bit = 599; BMR = 1,336%	44
Gambar 4.20 Hasil Ekstraksi Bit algoritma Mathur, sinyal Tone 5 kHz bervariasi, bit = 565; BMR = 7,788%	44
Gambar 4.21 Hasil Ekstraksi Bit algoritma Mathur, sinyal DTMF, bit = 409; BMR = 34,963%	45
Gambar 4.22 Hasil Ekstraksi Bit algoritma Mathur, sinyal White Noise, bit = 317; BMR = 40,063%	45
Gambar 4.23 Diagram Batang Hasil Perbandingan BMR.....	46

DAFTAR TABEL

Tabel 3.1 Sinyal-sinyal yang akan digunakan	19
Tabel 4.1 Rekapitulasi Seluruh Pengujian yang dilakukan	38
Tabel 4.2 Hasil Perbandingan Bit Mismatch Rate	46

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar belakang

Underwater Acoustics Communication System (UACS) adalah sistem telekomunikasi yang digunakan pada media bawah air. UACS telah banyak berperan dalam perkembangan teknologi dan informasi kemaritiman. Sistem ini menggunakan sinyal akustik atau suara sebagai media pengirim informasi, tidak seperti sistem telekomunikasi pada umumnya yang menggunakan gelombang radio. Sinyal akustik dipilih karena kemampuannya merambat dalam lingkungan bawah air yang lebih baik daripada jenis sinyal lainnya.

Dalam sistem telekomunikasi, kerahasiaan informasi terkadang menjadi kebutuhan penting dalam rangka menjaga keamanan dan integritas sebuah konten informasi. Proses pengamanan suatu informasi tersebut bisa menggunakan proses enkripsi ataupun digital signature. Enkripsi adalah proses mengamankan suatu informasi yang mengubah pesan asli (plaintext) menjadi pesan tersandi (ciphertext) dengan suatu algoritma yang diberi kunci khusus (secret key), sedang Digital Signature merupakan penanda pada data (informasi) yang memastikan bahwa data tersebut sesuai aslinya (tidak ada yang berubah). Dengan begitu, Digital Signature dapat memenuhi setidaknya dua syarat keamanan jaringan, yaitu Authenticity dan Nonrepudiation.

Pesan yang terenkripsi dapat diubah kembali menjadi seperti semula dengan proses dekripsi, dimana ciphertext akan diproses kembali menggunakan secret key yang sama. Sementara untuk membuka Digital Signature diperlukan kunci privat tertentu. Setelah dibuka menggunakan kunci tersebut, maka harus menghasilkan informasi yang sama dengan aslinya. Jika tidak, maka data (informasi) tersebut sudah mengalami perubahan. Ini merupakan salah satu syarat keamanan jaringan, dimana teknik autentifikasi diperlukan untuk menjamin keaslian data dari perubahan yang dilakukan pihak lain. Dan bila hasil Digital Signature cocok dengan kunci privat yang dipegang oleh pihak penerima, maka dapat dipastikan bahwa pengirim adalah pemegang kunci privat yang sah. Dengan demikian, Digital Signature telah menjalankan tugasnya dalam keamanan jaringan.

Di sisi lain penggunaan UACS yang semakin luas juga menuntut peningkatan keamanan (security) sistem, baik dari sisi konten informasi maupun dari keamanan jaringannya sendiri. Pada tugas akhir ini akan

dibahas salah satu metode pembangkitan secret key berbasis Received Signal Strength (RSS) untuk keamanan UACS. Pembahasan direncanakan terdiri dari simulasi sistem (menggunakan program Matlab) dan implementasinya dengan menggunakan peralatan seperti hydrophone, mixer, amplifier, dan speaker underwater untuk membentuk sistem UACS sederhana.

1.2 Rumusan Masalah

Adapun permasalahan yang akan dibahas dalam tugas akhir ini meliputi:

1. Bagaimana merekonstruksi UACS menggunakan perangkat-perangkat sederhana yang dapat menggambarkan sistem komunikasi bawah air.
2. Bagaimana mengimplementasikan pembangkitan *secret key* berbasis RSS pada sistem komunikasi bawah air.
3. Bagaimana melakukan pengukuran yang dapat mencerminkan keberhasilan sistem komunikasi bawah air.

1.3 Batasan Masalah

Dalam implementasi pembangkitan *Secret Key* berbasis RSS dalam tugas akhir ini, lingkup pembahasannya dibatasi oleh hal-hal sebagai berikut:

1. Perhitungan teoritis dilakukan menggunakan program Matlab.
2. Simulasi pembangkitan Secret Key dibuat juga menggunakan program Matlab.
3. Miniatur Sistem Komunikasi Bawah Air dibuat menggunakan komponen hydrophone, mixer, amplifier, dan speaker underwater.
4. Pengukuran realisasi sistem komunikasi bawah air dilakukan di kolam Hidrodinamika ITS.

1.4 Tujuan

Tujuan Tugas Akhir ini adalah untuk :

1. Merealisasikan sistem komunikasi bawah air (UACS) menggunakan perangkat-perangkat sederhana (hydrophone, mixer, amplifier, speaker underwater, dll).
2. Mengimplementasikan metode kriptografi untuk keamanan sistem komunikasi bawah air.
3. Mengaplikasikan metode pembangkitan *secret key* berbasis RSS pada sistem komunikasi bawah air.

1.5 Metodologi

Metodologi yang diterapkan terdiri dari tahapan-tahapan berikut :

1. Studi literatur

Mencari dan mempelajari beberapa literatur (buku, paper, website, dll) yang akan digunakan sebagai bahan kajian dan penulisan buku tugas akhir.

2. Perumusan Masalah dan Batasannya

Pada tahapan ini dilakukan identifikasi berbagai permasalahan yang akan diselesaikan dalam tugas akhir.

3. Perancangan Sistem

Mendesain blok diagram sistem komunikasi bawah air, pembuatan program pembangkitan *secret key*, dan pengujian menggunakan tools (software & hardware).

4. Pembuatan Simulasi Sistem Riil

Merangkai beberapa perangkat menjadi sistem untuk membangkitkan *secret key* berbasis RSS.

5. Pengujian Sistem di Lapangan

Membuat sistem komunikasi bawah air secara sederhana yang menerapkan unsur Autentifikasi untuk menunjang keamanan jaringan.

6. Analisa dan Kesimpulan

Melakukan analisa semua data dan hasil pengukuran yang diperoleh, untuk selanjutnya diambil kesimpulan.

7. Penulisan Buku Tugas Akhir

Dalam tahap ini dilakukan penulisan buku laporan tugas akhir.

1.6 Sistematika

Pembahasan Tugas Akhir ini dibagi menjadi lima bab dengan sistematika sebagai berikut :

Bab I Pendahuluan

Pada bab ini dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika pembahasan, dan relevansi.

Bab II Teori Penunjang

Pada bab ini dijelaskan mengenai tinjauan pustaka yang mendukung dalam pengimplementasian pembangkitan *secret key* berbasis RSS pada UACS.

Bab III Rancangan Implementasi Pembangkitan *Secret Key* berbasis RSS pada Sistem Komunikasi Bawah Air

Pada bab ini dijelaskan simulasi dan implementasi kompresi data pada peralatan UACS.

Bab IV Analisa Data

Bab ini berisi data pengujian dari sistem yang dibangun dimana data yang didapat dianalisa untuk mempermudah menarik kesimpulan.

Bab V Penutup

Bab ini berisi kesimpulan dan saran

1.7 Relevansi / Manfaat

Dengan adanya tugas akhir ini, diharapkan dapat memberikan referensi dalam menerapkan teknik kompresi data pada UACS, dan dikemudian hari dapat dikembangkan untuk kebermanfaatan orang banyak.

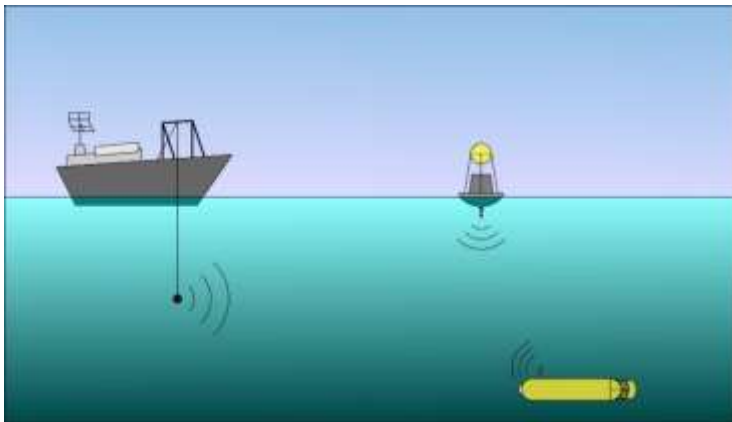
BAB II

TEORI PENUNJANG

2.1 Sistem Komunikasi Akustik Bawah Air

Sistem Komunikasi Akustik Bawah Air adalah sekumpulan perangkat komunikasi yang menggunakan suatu teknik transmisi data melalui media bawah air. Komunikasi bawah air sulit karena keterbatasan kemampuan sinyal yang dipakai, yaitu sinyal akustik. Faktor yang mempengaruhi yaitu propagasi *multi-path*, variasi waktu dari kanal, bandwidth yang kecil, dan atenuasi sinyal yang tinggi, khususnya untuk jarak jauh.

Karakteristik kanal dipermukaan bumi dan bawah air tentu sangat berbeda. Kondisi kanal bawah air dipengaruhi oleh temperatur air, kepekatan, dan tekanan. Pada komunikasi bawah air, sinyal yang digunakan adalah sinyal akustik dengan frekuensi berkisar antara 10 Hz sampai 1 MHz. Sinyal akustik merambat ke berbagai arah, yang menyebabkan sinyal tersebut akan dipantulkan oleh permukaan ataupun dasar air dan menyebabkan *multipath*. *Path-path* tersebut adalah *direct path*, *surface duct*, *bottom bounce*, *convergence zone*, *deep sound channel*, dan *reliable acoustic path*. Kecepatan sinyal akustik bergantung pada suhu, salinitas, dan tekanan air. Selain itu, Pada kanal bawah air terdapat Absorption Loss yang disebabkan oleh kedalaman, suhu, salinitas, serta keasaman air. Kanal bawah airpun dapat terkena noise dari



Gambar 2.1 Sistem Komunikasi Bawah Air

aktivitas-aktivitas bawah air seperti aktivitas vulkanik, seismik, aktivitas kapal, turbulensi, dan lain lain.

2.2 Propagasi Gelombang Akustik Bawah Air

Propagasi akustik bawah air bergantung pada banyak faktor, salah satunya adalah penurunan intensitas sinyal, yang disebabkan oleh persebaran geometri pada satu sisi dan absorpsi (penyerapan) pada sisi lainnya. Hal ini berkaitan dengan susunan kimiawi dari air, dan adalah salah satu fenomena yang pasti ada saat melakukan propagasi bawah air yang membatasi frekuensi kerja pada kanal bawah air. Mengestimasi kerugian propagasi merupakan proses penting dalam evaluasi kinerja suatu sistem komunikasi bawah air. [1]

Pada kanal bawah air kecepatan gelombang akustik mendekati 1500 m/s (diantara 1450 m/s dan 1550 m/s, tergantung pada tekanan, salinitas, dan suhu). Angka ini didapat dari persamaan:

$$c = \sqrt{\frac{E}{\rho}} = \sqrt{\frac{1}{\rho\chi}} \quad (2.1)$$

Dimana:

c = cepat rambat gelombang akustik (m/s)

E = modulus elastis (N/m²)

(jika berbentuk cair nilai $E = 1/\chi$, dimana E adalah nilai kompresabilitas)

ρ = massa jenis (kg/m³)

Sinyal akustik pada umumnya merupakan getaran yang teratur. Hal ini dicirikan dengan frekuensi f atau dengan periode T . Frekuensi yang digunakan pada akustik bawah air berkisar antara 10 Hz sampai 1 MHz, tergantung aplikasi.

Panjang gelombang adalah hubungan antara ruang dengan periode, dimana merupakan jarak antara dua titik pada medium yang mengalami getaran dengan jeda waktu T , atau pergeseran fasa sebesar 2π . Dengan kata lain, jarak yang ditempuh gelombang selama satu periode sinyal dengan kecepatan c . Pernyataan tersebut dibuktikan oleh persamaan berikut:

$$\lambda = cT = \frac{c}{f} \quad (2.2)$$

Dimana:

λ = panjang gelombang (m)

T = periode sinyal (s)

f = frekuensi (Hz).

Untuk cepat rambat suara sebesar 1500 m/s, panjang gelombang akustik bawah air akan bernilai 150m pada frekuensi 10 Hz, 1,5 m pada 1 kHz, dan 0,0015 m pada 1 MHz. Nilai-nilai frekuensi dan panjang gelombang yang bervariasi ini tentunya berkaitan dengan banyak proses fisik, baik untuk propagasi gelombang di dalam air, dan untuk karakteristik dari sistem akustik itu sendiri. Batasan utama penggunaan frekuensi untuk aplikasi tertentu adalah:

- Atenuasi gelombang akustik di dalam air, membatasi jangkauan maksimal yang dapat digunakan, yang mana efeknya meningkat dengan cepat dengan peningkatan frekuensi;
- Dimensi dari sumber gelombang, yang meningkat pada frekuensi rendah, untuk daya transmisi tertentu;
- Pemilihan spasial yang berhubungan dengan pengarahan dari sumber dan penerima akustik, membaik (untuk ukuran transduser tertentu) searah dengan kenaikan frekuensi;
- Target respon akustik, tergantung pada frekuensi; sebuah target akan memantulkan daya lebih sedikit jika dimensinya lebih kecil sehubungan dengan panjang gelombang akustik.

Salah satu pertimbangan pada perhitungan propagasi bawah air adalah rugi-rugi. Ada beberapa penyebab rugi-rugi dalam propagasi bawah air, yaitu akibat persebaran geometri, akibat penyerapan, akibat propagasi konvensional, dan akibat efek gelembung udara.

2.2.1 Pengaruh Persebaran Geometri

Propagasi suatu gelombang akustik dari suatu sumber menyebarkan energi ke segala arah yang semakin lama semakin membesar. Semakin luas penyebarannya, semakin kecil intensitas energi dari gelombang akustiknya. Proses ini disebut dengan rugi-rugi persebaran geometri.

Rugi rugi transmisi dari persebaran geometri dapat dihitung dengan persamaan:

$$TL = 20 \log\left(\frac{R}{R_{1m}}\right) \quad (2.3)$$

Dimana:

TL = Rugi-rugi transmisi (dB)

R = jarak yang diuji (m)

R_{1m} = jarak referensi (m).

Rugi-rugi persebaran radial biasanya dinyatakan dengan rumus $TL = 20 \log R$, tanpa referensi; meskipun tidak benar, penggunaan ini lebih mudah dan lebih sering digunakan.

2.2.2 Dampak Penyerapan

Air laut adalah media propagasi yang bersifat disipatif; air laut menyerap sebagian energi dari gelombang yang ditransmisikan, yang mana terserap melalui viskositas atau reaksi kimia. Pengurangan amplitudo lokal sebanding dengan amplitudonya sendiri; oleh karena itu tekanan akustik berkurang secara eksponensial dengan jarak. Hal ini akan menambahkan kepada rugi-rugi persebaran.

Penyerapan sering menjadi faktor utama yang membatasi pada propagasi akustik. Jumlahnya sangat tergantung pada media propagasi dan frekuensinya. Pada air laut, penyerapan berasal dari:

- Viskositas air murni, dimana efeknya berbanding lurus dengan kuadrat frekuensi
- Relaksasi molekul magnesium sulfat ($MgSO_4$) dibawah 100 kHz
- Relaksasi molekul Asam boric ($B(OH)_3$) dibawah 1 kHz.

Relaksasi molekular terdiri pada disosiasi dari senyawa ion pada larutan (disini $MgSO_4$ dan $B(OH)_3$), dikarenakan variasi tekanan lokal yang disebabkan oleh gelombang akustik. Proses ini berpengaruh besar terhadap penyerapan suara oleh air laut. Jika periode tekanan lokal lebih panjang dari waktu yang dibutuhkan molekul untuk memperbaiki dirinya (waktu relaksasi), proses ini akan berulang terus menerus, dan menyerap energi secara permanen. Jadi atenuasi akibat proses ini muncul pada frekuensi yang lebih rendah daripada frekuensi relaksasi karakteristik dari senyawa terkait.

Sejak awal dari akustik bawah air, banyak perhatian ditujukan pada pemodelan koefisien penyerapan, dan banyak model telah diajukan. Salah satu model terbaru adalah sebagai berikut:

$$\alpha = C_1 \frac{f_1 f^2}{f_1^2 + f^2} + C_2 \frac{f_2 f^2}{f_2^2 + f^2} + C_3 f^2 \quad (2.4)$$

Dimana:

- α = koefisien penyerapan (dB/km)
- f_i = frekuensi relaksasi (Hz)
- C_i = koefisien.

Dua variabel pertama pada persamaan tersebut menunjukkan kontribusi dari dua proses relaksasi; variabel ketiga terhubung ke viskositas air murni. Frekuensi relaksasi f_i dan koefisien C_i tergantung pada suhu, tekanan hidrostatik, dan salinitas; mereka ditentukan oleh eksperimen di laboratorium atau di laut.

2.2.3 Rugi-rugi Propagasi

Atenuasi secara sistematis digunakan sebagai pendekatan pertama saat mengevaluasi rugi-rugi propagasi dan kinerja dari sistem akustik bawah air. Rugi-rugi transmisi merupakan akumulasi dari rugi-rugi akibat persebaran geometri serta dampak penyerapan air yang melemahkan sinyal yang menjalar di dalamnya, sehingga persamaannya dapat dituliskan sebagai berikut:

$$TL = 20 \log \left(\frac{R}{R_{1m}} \right) + \alpha R \quad (2.5)$$

Namun, pada penggunaan umumnya digunakan persamaan yang lebih sederhana berikut (meski secara formal salah):

$$TL = 20 \log(R) + \alpha R \quad (2.6)$$

Untuk sistem yang menggunakan *echo* (pantulan dari target), maka akan mengalami rugi-rugi propagasi dua kali lipat yaitu pada saat mengirim dan kembalinya sinyal, sehingga persamaannya menjadi:

$$2TL = 40 \log(R) + 2\alpha R \quad (2.7)$$

2.2.4 Efek Gelembung Udara

Gelembung udara pada air laut umumnya disebabkan oleh pergerakan permukaan laut dan juga oleh perpindahan badan kapal di dalam air. Mereka membentuk suatu lapisan yang dekat dengan permukaan, dimana pembauran dua fase (air dan udara) merubah karakteristik akustik media propagasi di sekitarnya. Proses ini berkurang skalanya seiring dengan bertambahnya kedalaman, karena tekanan hidrostatik menahan gelembung, dan karena proses pembentukan gelembung udara terjadi pada daerah dekat permukaan laut (gelombang dan badan kapal). Pada area lebih dari kedalaman 10 m – 20 m, efek ini dapat diabaikan.

Gangguan setempat dari media propagasi memiliki dampak yang besar untuk kinerja SONAR pada badan kapal. Gelembung udara menimbulkan efek berupa:

- Atenuasi tambahan, ditambahkan pada efek penyerapan, menyebabkan pelemahan sinyal yang ditransmisikan dan menutupi transduser penerima;
- Merubah kecepatan gelombang akustik lokal, menyebabkan pantulan di dalam lapisan permukaan;
- Hamburan gema gelombang parasit, yang biasanya dapat diamati, sebagai contoh pada saat awal sinyal direkam oleh *echosounder*.

Proses ini dapat sangat mempengaruhi kinerja sensor akustik, yang dapat menyebabkan kerusakan, seperti yang umum dijumpai di lapangan. Berdasarkan ukuran dan kerapatan jumlahnya, gelembung dapat mengurangi jumlah pantulan, tetapi juga dapat mengurangi kualitas pengukuran, yang pada akhirnya menimbulkan kehilangan deteksi atau gema parasit.

Tentunya pengendalian parameter sebanyak mungkin pada proses pengganggu kinerja ini akan sangat membantu. Salah satu faktor penentu keberadaan gelembung adalah lingkungan sekitar, yang mana tidak dapat dikontrol (seperti pergerakan permukaan laut karena cuaca, aktivitas biologi, pergerakan kapal, dan lain-lain). Selain faktor tersebut, bentuk badan kapal, kecepatan tempat instrumen, serta posisi dan dimensi dari transduser adalah parameter penting yang harus dikontrol. Khususnya, ketika meneliti desain dan instalasi antenna sonar pada badan kapal, harus ada perlakuan khusus dengan menempatkannya jauh dari daerah permukaan air dangkal yang berpotensi mengandung gelembung. Hal ini dicapai dengan penelitian karakteristik hidrodinamis dari badan kapal, melalui berbagai simulasi numerik dan eksperimen, yang mengarah pada instalasi transduser pada bagian depan badan kapal.

Gelembung permukaan juga merubah karakteristik pantulan gelombang akustik pada permukaan laut, yang ditutupi oleh sebuah lapisan penyerapan. Efek atenuasi tambahan ini perlu diperhatikan, ketika memodelkan sinyal sonar yang merambat pada banyak kanal dan terpantul pada permukaan.

Telah banyak teori dan eksperimen tentang gelombang udara ini dan karakteristiknya. Setiap gelombang udara bertindak sebagai penghalang dengan impedansi akustik yang tinggi, dan karenanya merupakan penyebar yang kuat dari gelombang akustik.

Adanya efek penyebaran geometri dalam air akan menimbulkan efek resonansi yang besarnya dapat dihitung dengan rumus pendekatan sebagai berikut:

$$f_R = \frac{3.25}{a} \sqrt{1 + 0.1z} \quad (2.8)$$

Dimana:

f_R = frekuensi resonansi (Hz)
 z = kedalaman (m)
 a = radius (m)

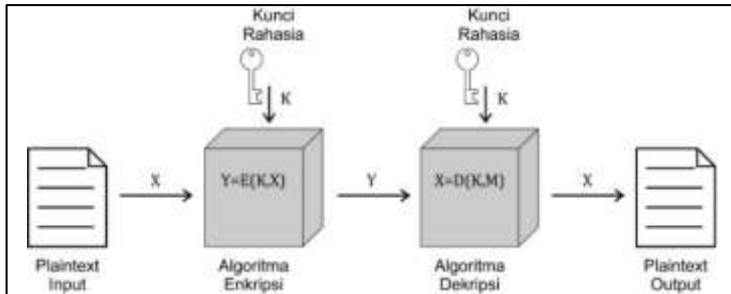
Penyerapan *visco-thermal* oleh dinding gelembung menambah efek persebaran. Ketika populasi gelembung meningkat, intensitas dari transmisi gelombang akustik dilemahkan dengan jumlah proses persebaran (sebagian besar berasal dari resonansi gelembung) dan proses penyerapan. Rugi-rugi dari intensitas transmisi dapat dimodelkan dengan koefisien penyerapan ekuivalen lokal, tergantung dari masing-masing karakteristik dari gelembung dan distribusi ukuran statistiknya.

2.3 Kriptografi

Kriptografi adalah disiplin ilmu yang mempelajari tentang teknik-teknik keamanan komunikasi (informasi, jaringan, server, dll.) untuk melindungi dari tindak kejahatan oleh pihak ketiga. Secara umum, kriptografi membahas tentang konstruksi dan analisis protokol sedemikian rupa sehingga pihak ketiga atau publik tidak dapat membaca pesan rahasia, memasuki jaringan tertentu, ataupun mengakses pusat data yang dilarang.

Sebenarnya ada bermacam-macam aspek pada keamanan data (informasi), mulai dari unsur kerahasiaan (*confidentiality*), integritas data (*integrity*), dan ketersediaan data saat dibutuhkan bagi pihak-pihak yang memiliki ijin akses terhadap informasi tersebut. Sementara salah satu usaha untuk menjaga integritas (akurasi) data agar tidak berubah dari aslinya antara lain menggunakan teknik kriptografi.

Kriptografi sesungguhnya berasal dari konsep-konsep matematika dan algoritma untuk mengubah suatu pesan menjadi pesan lain yang sulit dipahami. Algoritma ini digunakan untuk pembuatan kunci kriptografi, verifikasi digital, berselancar di internet, dan komunikasi rahasia.



Gambar 2.2 Skema Enkripsi Sederhana

2.3.1 Enkripsi

Salah satu komponen terpenting dalam kriptografi adalah enkripsi. Enkripsi adalah proses mengamankan suatu informasi dengan cara mengubah informasi tersebut menjadi suatu deretan bit acak dengan suatu algoritma tertentu. [2]

Enkripsi dilakukan dengan tujuan menjaga informasi agar tidak bisa dibaca oleh pihak yang tidak berkepentingan, sedangkan untuk mengembalikan menjadi seperti semula digunakan teknik dekripsi. Dalam hal proses enkripsi-dekripsi, algoritma yang digunakan harus sama.

Sebagai gambaran, implementasi enkripsi pada suatu data yang akan ditransmisikan dapat dilihat pada gambar 3.2. Dalam skema tersebut, data yang masih asli (plaintext) akan diacak menggunakan algoritma enkripsi tertentu (proses enkripsi dengan *secret key*) sebelum dikirim ke tujuan. Setelah sampai di tujuan, data yang masih dalam kondisi terenkripsi (teracak) harus dikembalikan seperti aslinya lewat proses dekripsi, menggunakan *secret key* yang sama dengan pengiriman.

2.3.2 Kunci

Kunci kriptografi (*secret key*) adalah deretan bit yang digunakan oleh algoritma untuk proses enkripsi maupun dekripsi. Kunci ini bersifat pribadi untuk memastikan komunikasi yang aman.

Dalam mendesain sistem keamanan, adalah bijak untuk mengasumsikan bahwa pihak penyerang sudah mengetahui algoritma kriptografinya. Kunci seringkali lebih mudah dilindungi karena ukurannya yang kecil dan mudah untuk diganti.

Kunci merupakan data yang bersifat sensitif yang harus dijaga kerahasiaannya, oleh karena itu dibutuhkan proses pembuatan dan distribusi kunci yang aman. Salah satunya adalah dengan membuat dan mendistribusikannya melalui jalur fisik, dimana pengiriman kunci dilakukan secara fisik ke dua pihak yang akan berkomunikasi.

2.3.3 Pembangkitan Kunci

Untuk tetap memenuhi unsur kerahasiaan (*confidentiality*), kunci (*secret key*) biasanya hanya dipakai sampai batas waktu tertentu. Kunci yang demikian disebut dengan *session key*. Karena kunci harus berganti setiap waktu (*session key*), maka harus ada proses pembangkitan kunci yang baru tiap waktu tertentu dalam rangka menjaga agar komunikasi tetap aman.

Proses pembangkitan kunci dapat dilakukan pada lapisan fisik dengan memanfaatkan karakteristik kanal komunikasi. Salah satu karakteristik kanal yang dapat digunakan sebagai basis pembangkitan kunci (*secret key*) adalah besaran kuat sinyal yang diterima atau *Received Signal Strength* (RSS).

Teknik pembangkitan kunci berbasis RSS memiliki kriteria (karakteristik) yang unik dan berbeda nilainya pada tiap kanal, sehingga sulit untuk meniru satu kanal dengan kanal lainnya. Kondisi inilah yang dimanfaatkan untuk menjaga privasi (kerahasiaan) selama komunikasi berlangsung.

2.4 Perangkat Hydrophone

Hydrophone adalah sebuah mikrofon yang khusus dibuat untuk pemakaian di dalam air (*underwater*). Prinsip kerja *hydrophone* sama dengan mikrofon pada umumnya, yaitu mengubah sinyal akustik/suara menjadi sinyal listrik, sehingga termasuk golongan *transducer*. Proses konversi sinyal ini dilakukan dengan memanfaatkan teknologi piezoelektrik, dimana energi listrik dihasilkan oleh material padat karena adanya tekanan mekanik. [3]

Meskipun ada beberapa jenis *hydrophone* berfungsi sebagai pengirim (*transmitter*), dalam sistem komunikasi bawah air (*underwater*), *hydrophone* biasanya difungsikan sebagai *receiver* untuk menerima sinyal akustik.



Gambar 2.3 Contoh Perangkat *Hydrophone*

Dalam pemakaian lain, *hydrophone* merupakan salah satu bagian terpenting dalam sistem SONAR (*Sound Navigation and Ranging*), teknik penjaran suara dalam air (mendeteksi keberadaan obyek lain dalam air), maupun sebagai alat navigasi. Dalam teknologi sonar sendiri, sudah mulai digunakan saat terjadi perang dunia pertama, dimana *hydrophone* dipakai untuk mendeteksi kapal-kapal selam di bawah laut. Pada masa ini, perkembangan teknologi sonar diperuntukkan/dikhususkan hanya pada bidang militer. Hingga pada akhir perang dunia kedua, barulah pengetahuan tentang akustik lebih berkembang luas. Bukan hanya untuk keperluan militer saja tapi juga digunakan untuk keperluan non-militer seperti pendidikan, penelitian, bahkan untuk komunikasi bawah air.

2.5 Speaker Bawah Air

Speaker bawah air dapat difungsikan sebagai pengirim (*transmitter*) sinyal akustik. Speaker bekerja dengan mengubah sinyal listrik menjadi sinyal suara/akustik, sehingga juga termasuk sebagai alat *transducer*. Sama seperti *hydrophone*, speaker juga salah satu komponen terpenting dalam SONAR.

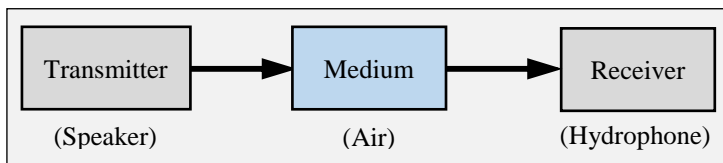


Gambar 2.4 Speaker Bawah Air

2.6 Ilustrasi Sistem Komunikasi Bawah Air

Secara sederhana, sistem komunikasi bawah air dapat dibangun dari beberapa alat yang sudah dijelaskan di atas (*hydrophone* dan speaker bawah air), dengan skema sebagai berikut.

Pada dasarnya, speaker yang berperan sebagai pemancar mengirimkan sinyal melalui suatu medium (dalam kasus ini air) untuk kemudian ditangkap oleh *hydrophone* sebagai penerima. Sinyal yang diterima lalu akan diolah oleh pihak penerima untuk dibangkitkan secret key.



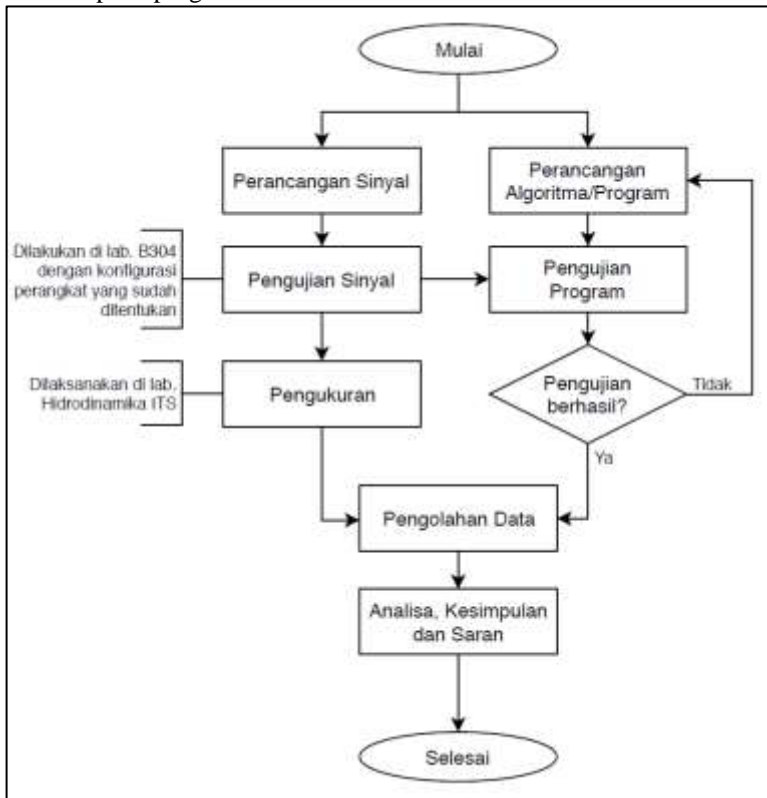
Gambar 2.5 Blok Diagram Sederhana Sistem Komunikasi Bawah Air

[halaman ini sengaja dikosongkan]

BAB III METODOLOGI PENELITIAN

3.1 Diagram Alur

Tugas akhir ini dikerjakan dengan beberapa tahapan proses. Proses penelitian implementasi pembangkitan secret key ini diawali dengan merancang algoritma pembangkitan secret key. Sebelum algoritma atau program yang dibuat bisa digunakan, terlebih dahulu dilakukan percobaan atau test pada program.



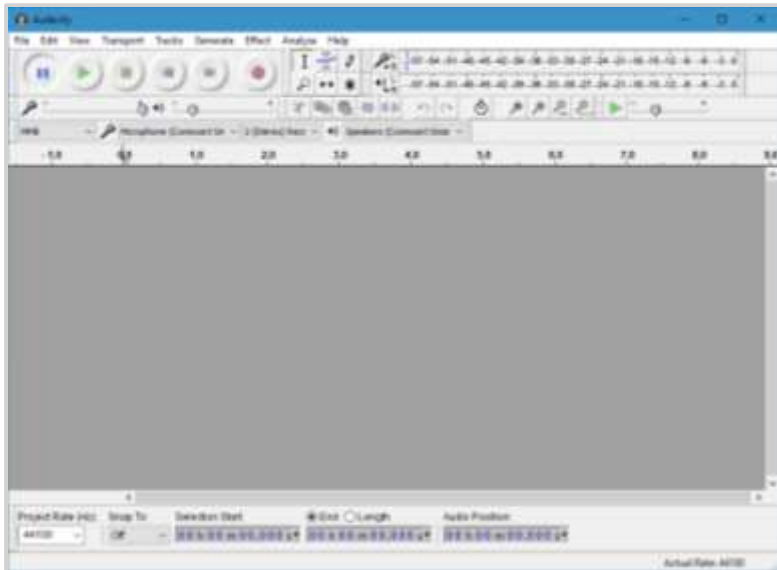
Gambar 3.1 Flowchart Penelitian

3.2 Perancangan Sinyal Akustik

Proses pembuatan sinyal akustik dilakukan dengan program Audacity. Aplikasi ini dapat digunakan untuk membuat maupun mengedit sinyal audio. Ada beberapa jenis sinyal yang dapat dibangkitkan, beberapa diantaranya yaitu tone, chirp, DTMF, dan noise. Berikut adalah interface dari Audacity.

Untuk membangkitkan sinyal, pilih “Generate” pada toolbar, lalu pilih jenis sinyal yang ingin dibangkitkan. Selanjutnya akan muncul window untuk mengatur spesifikasi sinyal yang akan dibuat, seperti frekuensi, jenis gelombang, dan durasi.

Ada 4 jenis sinyal yang akan digunakan pada penelitian ini, dengan salah satu jenis memiliki 3 variasi. Jenis sinyal yang digunakan adalah sinyal Chirp, Tone, DTMF, dan White Noise. Untuk lebih detailnya dapat dilihat pada **Tabel 3.1**.



Gambar 3.2 Interface Program Audacity

Tabel 3.1 Sinyal-sinyal yang akan digunakan

No	Jenis Sinyal	Durasi (detik)	Keterangan
1	Chirp	1,5	5 kHz – 12 kHz
2	Tone	1,5	5 kHz
3		1,4	5 kHz dan 8 kHz
4		1,5	5 kHz, variasi amplitude
5	DTMF	1,5	5947302, duty cycle 75%
6	Noise	1,5	White noise

3.3 Perancangan Algoritma Secret Key

Pada tugas akhir ini, ada dua jenis algoritma yang akan dipakai untuk implementasi pembangkitan *secret key*, yaitu menggunakan pendekatan Aono dan pendekatan Mathur. Kedua pendekatan tersebut memiliki beberapa persamaan dan perbedaan. Adapun persamaannya yaitu keduanya menggunakan pendekatan single-bit, dimana dari setiap nilai RSS akan dibangkitkan satu bit (0 atau 1) sesuai dengan besar nilai RSS-nya. Sedangkan keduanya memiliki perbedaan antara lain jumlah bit yang dibangkitkan dan batasan pembangkitan bit.

3.3.1 Pendekatan Aono

Pendekatan ini termasuk dalam kategori *single-bit*, dimana pada setiap nilai RSS tertentu akan dihasilkan satu bit. Pertama-tama, kedua pihak harus menentukan panjang kuncinya, yang dinotasikan dengan l . Setelah itu, nilai RSS dapat dikuantisasi dengan membagi jumlah bit sebesar masing-masing $l/2$ untuk nilai maksimum dan minimum RSS, dimana akan dibangkitkan bit “1” dan “0”. [4] Algoritma ekstraksi bit ini dapat dituliskan menggunakan program Matlab, seperti yang terlihat pada gambar 3.3.

Seperti yang ditunjukkan pada gambar 3.3, bahwa algoritma Aono memiliki komposisi yang mencerminkan proses pembangkitan *secret key* yang dimulai dari penentuan panjang kunci, batas range sinyal, pemilihan nilai RSS maksimum dan minimum berdasarkan parameter daya yang sampai di penerima (*receiver*). Proses tersebut berlangsung terus menerus hingga didapatkan jumlah bit *secret key* sebanyak yang diinginkan, dimana dimana jumlah separuhnya merupakan representasi dari jumlah bit 1 dan setengah sisanya mencerminkan jumlah bit 0 yang berhasil dibangkitkan.

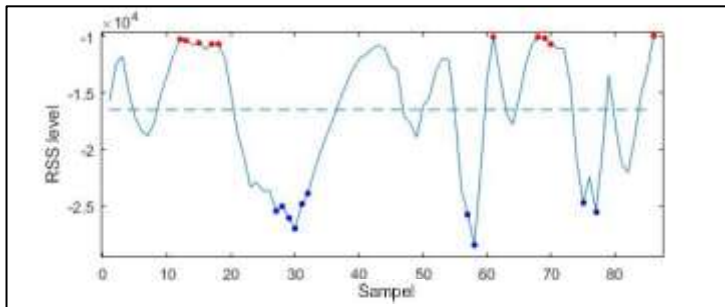
```

bitr1=zeros(size(RSS1));
key=128;
RSSmax1=RSS1;
RSSmin1=RSS1;
for i=1:key/2
    [Ma,Ia]=max(RSSmax1);
    bitr1(Ia)=2;
    RSSmax1(Ia)=avg1;
    [Mi,Ii]=min(RSSmin1);
    bitr1(Ii)=1;
    RSSmin1(Ii)=avg1;
end

bit1=[];
for i=1:length(bitr1)
    if bitr1(i) ~= 0
        bit1=[bit1 bitr1(i)];
    end
end
bit1=bit1 - 1;
bit1=bit1-1;

```

Gambar 3.3 Algoritma Aono



Gambar 3.4 Contoh pembangkitan bit dengan metode Aono

Dari algoritma pada **gambar 3.3**, dihasilkan nilai RSS seperti tertera pada **gambar 3.4**, bit-bit yang dihasilkan dari nilai-nilai RSS menempati nilai RSS dari yang paling besar untuk nilai “1” dan nilai RSS yang paling kecil untuk nilai “0”. Perlu diperhatikan bahwa jumlah bit “1” dan “0” adalah sama. Hal ini dikarenakan sebelum bit diekstraksi, perlu

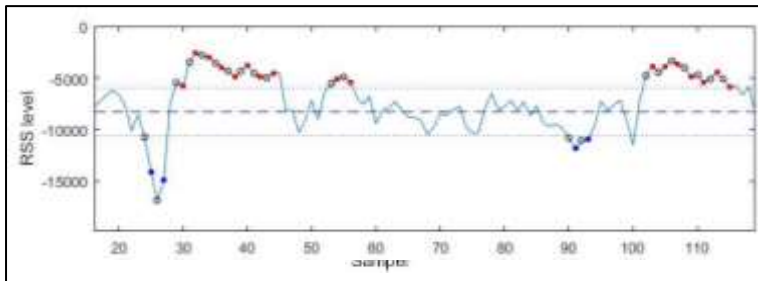
ditentukan dahulu jumlah bit yang akan dihasilkan. Jumlah masing-masing bit selalu bernilai sama, jadi penentuan awal jumlah bit harus bernilai genap.

3.3.2 Pendekatan Mathur

Pendekatan ini juga termasuk dalam kategori *single-bit*. Penentuan bit dipengaruhi oleh nilai q_+ dan q_- dari deret RSS-nya, dimana $q_{\pm} = \mu + \alpha \cdot \sigma$, α adalah quantizer level coefficient, μ adalah rata-rata dan σ adalah standar deviasi dari deret RSS. Selanjutnya akan dikuantisasi nilai RSS berturut-turut sebanyak n (dalam contoh ini, $n = 2$) yang berada diatas q_+

```
avg1 = sum(RSS1)/length(RSS1);
var1 = sum((RSS1-avg1).^2)/length(RSS1);
tau1 = sqrt(var1);
qup1 = avg1 + alfa*tau1;
qlw1 = avg1 - alfa*tau1;
k1 = length(RSS1);
i1=1;
bit1=[];
while i1 < k1
    if RSS1(i1) > qup1
        if RSS1(i1+1) > qup1
            bit1 = [bit1 1];
            i1=i1+2;
        else
            i1=i1+1;
        end
    elseif RSS1(i1) < qlw1
        if RSS1(i1+1) < qlw1
            bit1 = [bit1 0];
            i1=i1+2;
        else
            i1=i1+1;
        end
    else
        i1=i1+1;
    end
end
```

Gambar 3.5 Algoritma Mathur



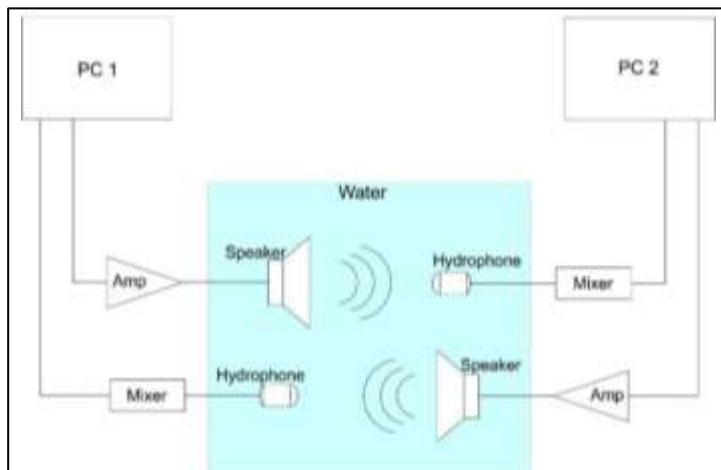
Gambar 3.6 Contoh pembangkitan bit dengan metode Mathur

dan dibawah q_- menjadi bit “1” dan “0”. [4] Pada program Matlab, dapat dituliskan algoritma Mathur seperti pada **gambar 3.5**.

Pada **gambar 3.6**, pembangkitan bit dipengaruhi oleh dua nilai, yaitu q_+ untuk bit “1” dan q_- untuk bit “0”. Proses ekstraksi bit tidak menggunakan masing-masing nilai RSS, tetapi dengan mengamati nilai RSS yang berurutan. Jika salah satu nilai RSS melebihi batas $q \pm$ dan nilai RSS berikutnya juga sama, maka akan dibangkitkan bit yang berkaitan.

3.4 Rancangan Sistem Komunikasi Bawah Air

Pada penelitian ini, diperlukan beberapa perangkat keras khusus yang dapat digunakan untuk pengukuran pada ekosistem bawah air.



Gambar 3.7 Rancangan Sistem Komunikasi Bawah Air

Perangkat keras yang dipakai adalah Speaker bawah air, hydrophone, Amplifier, Mixer, dan Laptop.

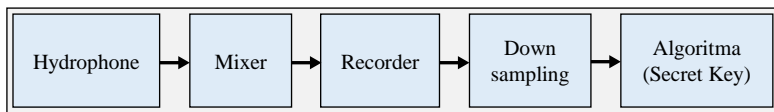
Pada proses pembangkitan *secret key*, diperlukan dua pihak yang akan berkomunikasi, pada kasus ini yaitu PC 1 dan PC 2. Proses pengiriman sinyal dilakukan secara bergantian (*half duplex*). Dari komputer pengirim, sinyal dilewatkan amplifier untuk dikuatkan terlebih dahulu sebelum nantinya akan dikonversi oleh speaker menjadi sinyal akustik. Dari sisi penerima, sinyal akustik yang diterima oleh *hydrophone* akan dikonversi menjadi sinyal digital yang akan dilewatkan ke mixer untuk dikuatkan (diberi gain) kembali. Terakhir, sinyal akan masuk ke komputer penerima untuk kemudian disimpan dan diolah. Proses ini akan diulang dengan mengganti komputer pengirim menjadi penerima dan sebaliknya. Ada enam jenis sinyal yang akan dipakai untuk pengiriman kedua pihak, yaitu sintal Chirp, Tone 5 kHz, Tone 5 & 8 kHz, Tone 5 kHz variasi, DTMF, dan white noise.

3.5 Perangkat Lunak Sistem Penerima

Selain perangkat keras, yang perlu dipersiapkan untuk membangkitkan secret key dari sistem penerima komunikasi bawah air adalah konfigurasi perangkat lunak yang digunakan. **Gambar 3.8** memaparkan blok diagram sistem penerima.

Pertama, sinyal diterima oleh *Hydrophone* yang terseambung ke Mixer. Pada mixer, sinyal akan dikuatkan (diberi gain) dengan perangkat keras mixer dan perangkat lunak Fast Track Ultra yang dapat mengatur besar gain yang digunakan. Dari mixer, sinyal kemudian ditangkap untuk kemudian disimpan oleh software recorder pada PC penerima dalam bentuk file audio (.wav). Sinyal audio kemudian akan diolah dengan algoritma downsampling untuk kemudian dibangkitkan secret key dengan algoritma pembangkitan yang sudah ditentukan.

Ada beberapa perangkat lunak yang digunakan dalam tugas akhir ini yaitu, Fast Track Ultra yang berfungsi menjembatani antara perangkat mixer dengan recorder. Software ini pada dasarnya berfungsi sebagai penguat sinyal. Sementara recordernya sendiri juga berupa perangkat lunak yaitu Adobe Audition CS6. Bila ingin diproses lebih lanjut,



Gambar 3.8 Blok Diagram Sistem Penerima

misalkan untuk dibangkitkan secret key-nya, maka output dari recorder bisa di-downsampling terlebih dahulu menggunakan program Matlab.

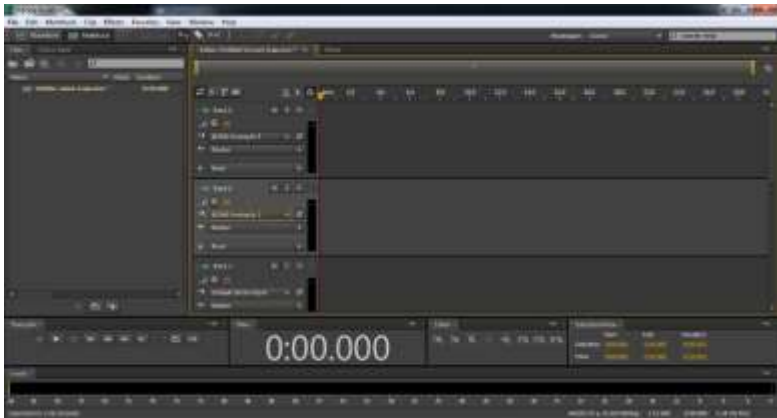
3.5.1 Fast Track Ultra

Pada tugas akhir ini akan digunakan Perangkat mixer M-Audio untuk menyambungkan hydrophone dengan PC penerima. Fast Track Ultra merupakan aplikasi atau driver dari M-audio yang mengindikasikan jika M-audio sudah tersambung dan siap digunakan.

Untuk mengkonfigurasi perangkat M-audio ini, hal pertama yang harus dilakukan adalah menginstall Fast Track Ultra. Tidak ada spesifikasi khusus untuk menginstall driver Fast Track Ultra, namun disarankan untuk komputer dengan OS Windows 7 atau kebawah. Kemudian setelah terinstall dengan benar, Jalankan perangkat lunak dan lakukan konfigurasi dengan mengatur parameter yang ada di panel software dan hardware. Pada window yang terlihat diatas dilakukan penyetelan dengan memperhatikan channel yang memberi respon.



Gambar 3.9 Pengaturan pada Fast Track Ultra



Gambar 3.10 Pengaturan Adobe Audition CS6

3.5.2 Adobe Audition CS6

Pada tugas akhir ini sinyal yang digunakan untuk mengirimkan informasi adalah sinyal akustik. Perangkat lunak Adobe Audition CS6 ini digunakan pada node penerima yang dalam hal ini adalah hydrophone. Perangkat lunak ini menerjemahkan hasil pengiriman sinyal akustik kedalam tipe data audio yang nantinya akan diolah di penerima guna menerjemahkan informasi.

Jika Fast Track Ultra menunjukkan respon yang diinginkan, saatnya beralih ke Adobe Audition CS6. Adobe Audition CS6 diatur dengan sample rate 44100 Hz, bit depth 16 bits, dan Master disetting Mono. Pada Pengaturan Audio hardware diatur pada class device ANSI. Selanjutnya Adobe CS6 siap untuk digunakan. Sebelum melakukan pengukuran pastikan pada window untuk mengaktifkan mode record dengan menekan tombol R pada window, setelah ditekan tombol dengan huruf R akan berwarna merah, dan Adobe Audition siap untuk merekam data. Untuk mengoperasikannya cukup dengan menekan tombol Rec atau kombinasi Shift + Space pada keyboard. Saat dilakukan pengukuran file akan tersimpan otomatis dengan format wav.

3.5.3 Program Downsampling

Downsampling adalah proses mengubah sample rate dari suatu sinyal. Proses downsampling dilakukan apabila sinyal yang telah tersimpan dalam recorder hendak diproses lebih lanjut.

```

RSS11=[];
ii=1;
while ii<(floor(length(RSS1)/dnsp)*dnsp)
    RSS11a=0;
    for iii=1:dnsp
        RSS11a=RSS11a+RSS1(ii);
        ii=ii+1;
    end
    RSS11=[RSS11 RSS11a/dnsp];
end

```

Gambar 3.11 Kode Matlab untuk algoritma downsampling



Gambar 3.12 Blok diagram downsampling

Dalam tugas akhir ini, sinyal yang tersimpan di dalam recorder akan diproses untuk membangkitkan secret key-nya. Karena tidak mungkin memproses seluruh sinyal, maka hanya beberapa bagian (*sample*) sinyal saja yang diambil berdasarkan proses pencuplikan (*downsampling*). Selanjutnya sinyal hasil cuplik itulah yang membangkitkan secret key-nya. Untuk keperluan tersebut, perangkat lunak untuk proses downsampling juga telah dibuat dalam tugas akhir ini dengan bahasa Matlab.

Dalam rancangan sistem komunikasi bawah air yang dibuat dalam tugas akhir ini, proses downsampling juga digunakan sebagai masukan sinyal dalam proses pembangkitan *secret key*. Dengan demikian, setelah dilakukan proses downsampling, hasilnya akan diproses menggunakan algoritma berbasis RSS untuk mempercepat proses pembangkitan kunci. Hanya saja, ada sedikit potensi kelemahan dalam proses downsampling ini karena sinyal hasil downsampling bisa agak berubah dari sinyal aslinya, sehingga dapat mengubah bit yang akan membangkitkan. Berikut adalah kode Matlab untuk proses downsampling.

Proses downsampling membagi jumlah sampel sinyal dengan sebuah bilangan bulat n . Setiap sampel dari suatu urutan m sampai dengan $m+n$ akan dirata-rata menjadi sampel baru.

Proses downsampling terjadi setelah sinyal disimpan oleh recorder. Sinyal hasil pengolahan algoritma downsampling kemudian

akan diolah untuk dibangkitkan secret key pada algoritma pembangkit secret key yang telah ditentukan.

3.6 Skenario Pengujian

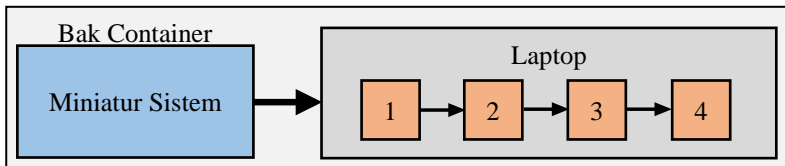
Ada dua fase pengujian terkait dengan pembangkitan secret key dalam tugas akhir ini, yaitu saat masih dalam proses rancangan sistem (menguji algoritma) dan pengukuran dalam sistem komunikasi bawah air secara riil.

3.6.1 Pengujian Algoritma

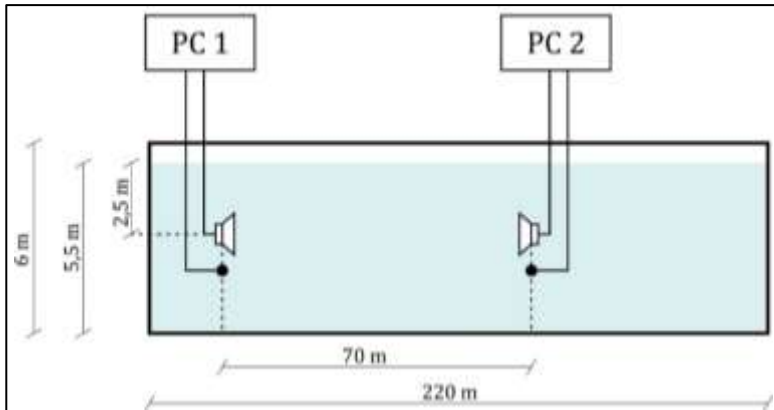
Tujuan pengujian pada tahap ini untuk melihat apakah algoritma pembangkitan secret key yang telah dibuat bisa berfungsi atau tidak. Untuk itu dibuatlah miniatur sistem komunikasi bawah air namun melibatkan seluruh komponen yang nantinya dipakai pada pengujian riil. Adapun skema pengujian algoritma yang telah dilakukan dalam tugas akhir ini tersusun seperti dalam **gambar 3.13**, dengan sinyal masukan bervariasi seperti yang tercantum dalam table 3.1

Untuk menguji algoritma secret key digunakan langkah-langkah sebagai berikut:

1. Menyusun rangkaian seperti gambar di atas
2. Aktifkan recorder
3. Pilih salah satu sinyal seperti dalam tabel 3.1
4. Jalankan sinyal pada speaker
5. Rekam sinyal yang sampai di penerima dengan recorder
6. Masukkan sinyal rekaman pada program downsampling
7. Gunakan Algoritma Aono
8. Ulangi langkah 1-7 dengan input sinyal lain dari tabel 3.1
9. Ulangi langkah 1-8 dengan mengganti algoritma Aono dengan algoritma Mathur
10. Ulangi langkah 1-9, dengan membalik lokasi pengirim sebagai penerima dan lokasi penerima sebagai pengirim



Gambar 3.13 Proses Pengujian Algoritma, (1) Recorder, (2) Downsampling, (3) Algoritma secret key, (4) Bit secret key terbangkitkan



Gambar 3.13 Skema Pengukuran Sistem Komunikasi Bawah Air

3.6.2 Pengukuran di lapangan

Langkah-langkah pengukuran ini sebenarnya mirip dengan tahapan pengukuran algoritma. Yang membedakan dari keduanya adalah miniatur sistem komunikasi bawah air yang semula menggunakan bak container diganti dengan kolam Hidrodinamika ITS yang berukuran 200 m x 11 m x 5,5 m. Pada gambar berikut akan dipaparkan detail pengukuran yang diadakan di kolam Hidrodinamika ITS.

Gambar 3.13 merupakan tampak samping dari skema pengukuran sebenarnya. Letak speaker dan hydrophone yang sebenarnya adalah sama yaitu pada kedalaman 2,5 meter.

BAB IV

ANALISA DATA DAN PEMBAHASAN

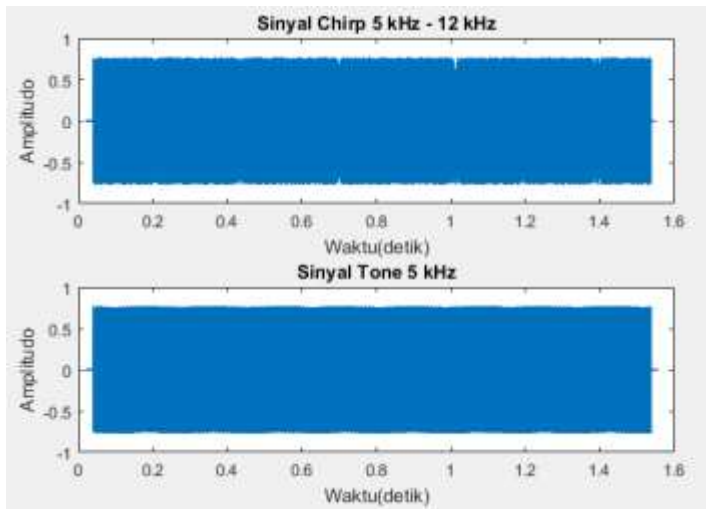
Pada bab ini akan dibahas tentang analisa dari implementasi pembangkitan secret key berbasis RSS. Dari beberapa jenis sinyal yang diuji dan dari dua jenis algoritma yang digunakan, akan dianalisa untuk mengetahui hasil terbaik. Hasil terbaik ditentukan oleh nilai Bit Mismatch Rate yang paling kecil.

4.1 Tampilan Hasil Pengujian Sinyal dan Program

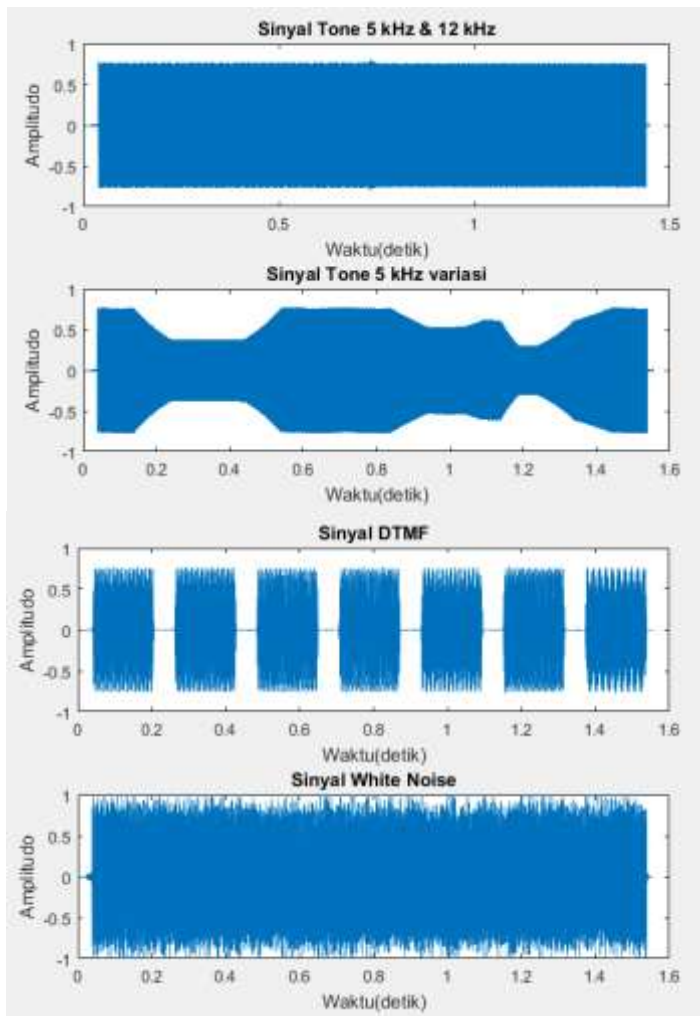
Pada tahap pertama dari pengujian pada tugas akhir ini, didapatkan beberapa hasil, diantaranya adalah sinyal yang telah dibangkitkan, pengujian sinyal di lab. B304, dan hasil pengujian pengolahan sinyal.

4.1.1 Hasil Pembangkitan Sinyal

Dengan bantuan software Audacity, dapat dibangkitkan sinyal-sinyal yang dibutuhkan untuk penelitian ini. Berikut adalah hasil pembangkitan dari enam sinyal:



Gambar 4.1 Hasil pembangkitan sinyal



Gambar 4.1 Hasil pembangkitan sinyal (lanjutan)

Bentuk tiga sinyal pertama pada gambar 4.1 di atas relatif sama, tetapi frekuensinya berbeda-beda, sehingga bentuk sinyal yang diterima akan relatif berbeda tiap-tiap sinyal.



Gambar 4.2 Konfigurasi pengujian Sinyal

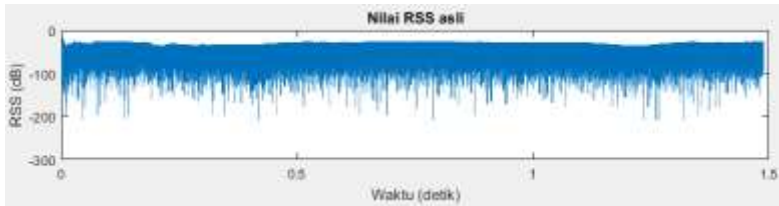
4.1.2 Realisasi Pengujian Sinyal

Dari gambar skema sistem komunikasi bawah air yang telah dijelaskan pada bab sebelumnya, selanjutnya dilakukan pengukuran di lab. B304 dengan tujuan untuk menguji sinyal yang telah dibangkitkan.

Pada konfigurasi pada **gambar 4.2**, perangkat *hydrophone* yang digunakan hanya satu buah. Maka untuk proses pengujian sinyal dilakukan secara bergantian. Begitu juga dengan perlengkapan amplifier, mixer dan laptop pengirim dan penerima. Satu laptop digunakan sebagai pengirim dan laptop lainnya digunakan sebagai penerima. Proses transmisi sinyal dilakukan bergantian dengan speaker kiri mengirim terlebih dahulu dan akan diterima oleh *hydrophone* di sebelah kanan. Selanjutnya, sambungan speaker akan diganti dengan speaker kanan dan *hydrophone* akan dipindah ke sebelah kiri untuk transmisi kedua.

4.1.3 Hasil Pengolahan RSS

Setelah semua sinyal dibangkitkan dan diuji, langkah selanjutnya adalah menghitung nilai RSS-nya. Perhitungan nilai RSS adalah hasil dari amplitudo sinyal yang dikuadratkan lalu diubah ke dalam bentuk desibel.



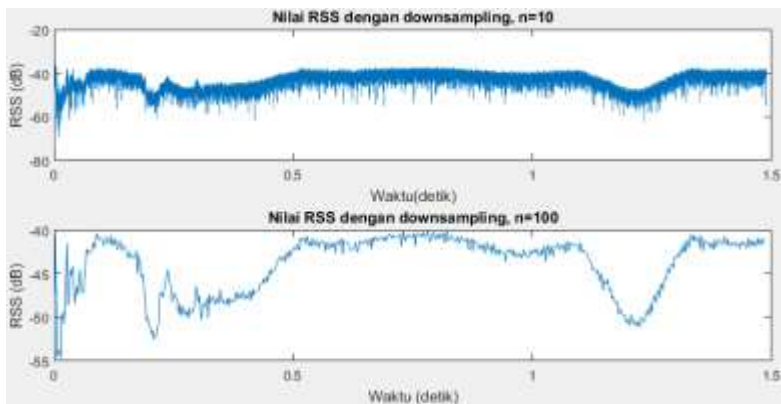
Gambar 4.3 Contoh Hasil RSS sinyal (Tone)

Gambar 4.3 adalah hasil perhitungan nilai RSS dari salah satu sinyal yaitu sinyal Tone 5 kHz var.

4.1.4 Hasil Downsampling

Proses downsampling dilakukan untuk mengurangi panjang sampel suatu sinyal. Pengurangan jumlah sampel ini juga dapat mempercepat proses pengolahan sinyal pada algoritma. **Gambar 4.4** merupakan hasil dari proses downsampling dari nilai RSS sinyal tone pada **gambar 4.3**.

Terdapat dua contoh downsampling pada gambar di atas, yaitu downsampling dengan pembagi $n = 10$ dan $n = 100$. Dari hasil tersebut, dapat dilihat bahwa panjang sampel berkurang menjadi $1/10$ dan $1/100$ dari panjang aslinya.



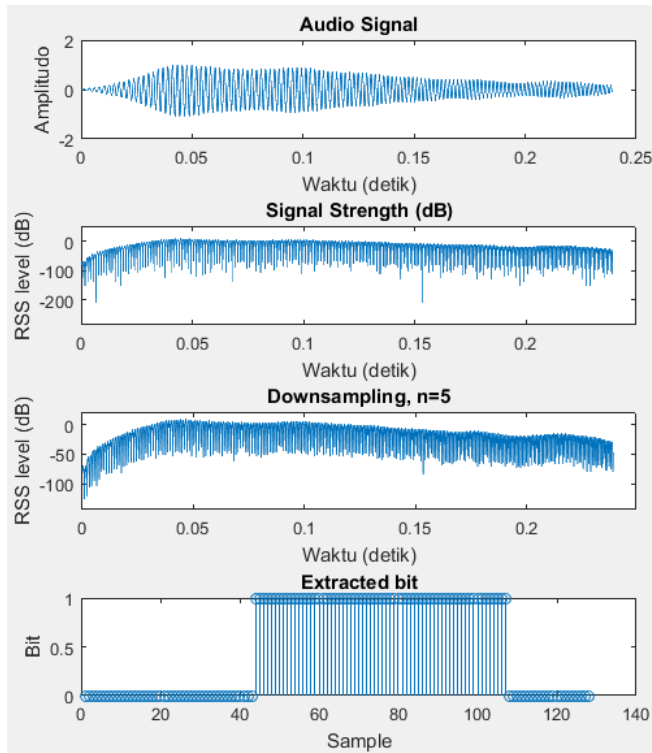
Gambar 4.4 Contoh hasil downsampling RSS sinyal (Tone)

4.2 Hasil Pengujian Algoritma

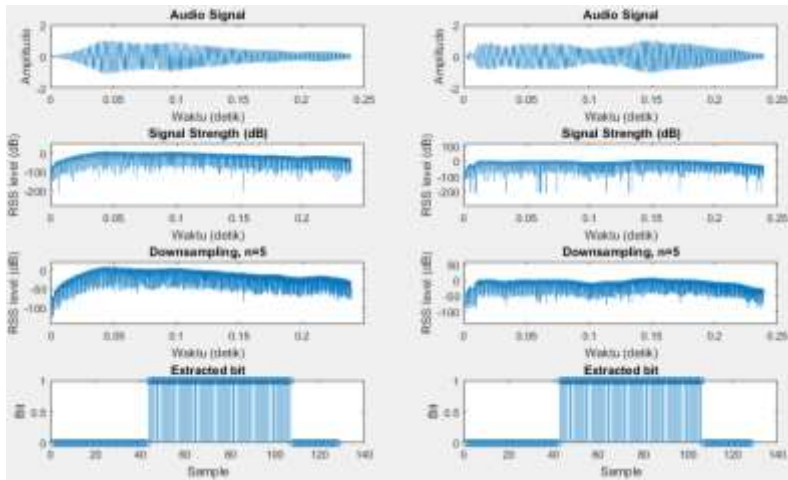
Setelah sinyal berhasil dibangkitkan, langkah selanjutnya adalah pengujian dari algoritma pembangkitan secret key yang telah dibuat. Kriteria kesuksesan pengujian algoritma adalah jika algoritma dapat mengolah sinyal dan dapat menghasilkan deretan bit secret key yang diinginkan.

4.2.1 Hasil Pengujian Algoritma Aono

Pada proses pengujian algoritma Aono, diasumsikan menggunakan masukan sinyal chirp, dengan downsampling $n=5$, dan panjang key 128 bit. Sehingga diperoleh hasil pengujian sebagai berikut.



Gambar 4.5 Contoh hasil ekstraksi bit dengan Algoritma Aono



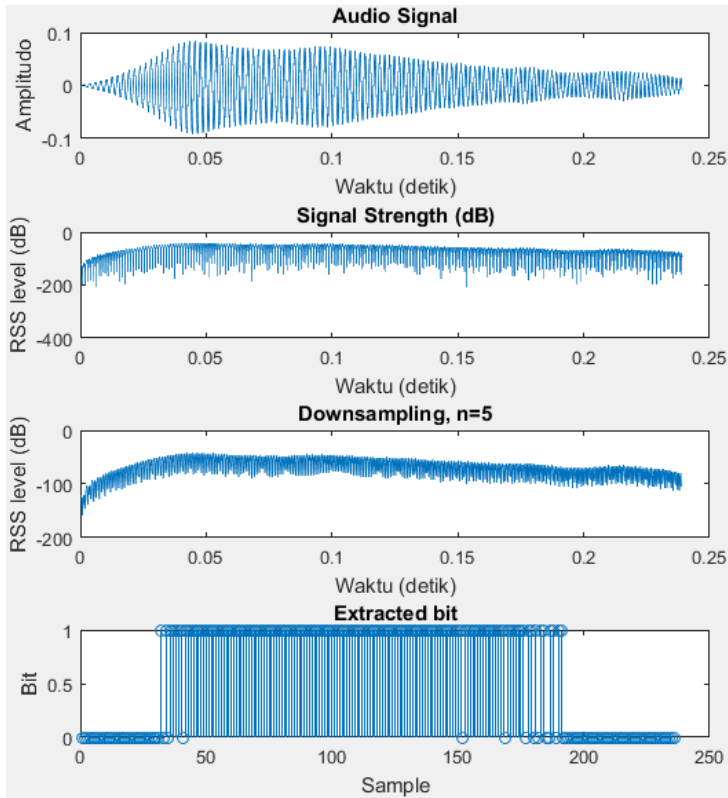
Gambar 4.6 Transmisi (kiri) A ke B dan (kanan) B ke A

Selanjutnya akan dilakukan transmisi sinyal kebalikannya, pihak pengirim (A) berfungsi sebagai penerima dan pihak penerima (B) berfungsi sebagai pengirim dan hasilnya bisa dilihat pada **gambar 4.6**.

Dari hasil perbandingan dua sinyal di atas, terlihat bahwa walaupun bentuk sinyal sedikit berbeda, namun bit yang dihasilkan terlihat sama antara dua sinyal. Tetapi nilai deret bit yang dihasilkan kurang bervariasi, dengan deretan nilai “0” pada awal dan akhir deret bit dan deretan nilai “1” pada pertengahan deret bit. Variasi nilai deret bit dipengaruhi baik oleh jenis sinyal yang dipakai dan juga algoritma yang dipakai.

4.2.2 Hasil Pengujian Algoritma Mathur

Parameter sinyal yang akan digunakan untuk pengujian algoritma Mathur hampir sama dengan parameter sinyal sebelumnya, yaitu sinyal chirp, dengan downsampling sinyal 1 $n=5$. Hasilnya adalah sebagai berikut.

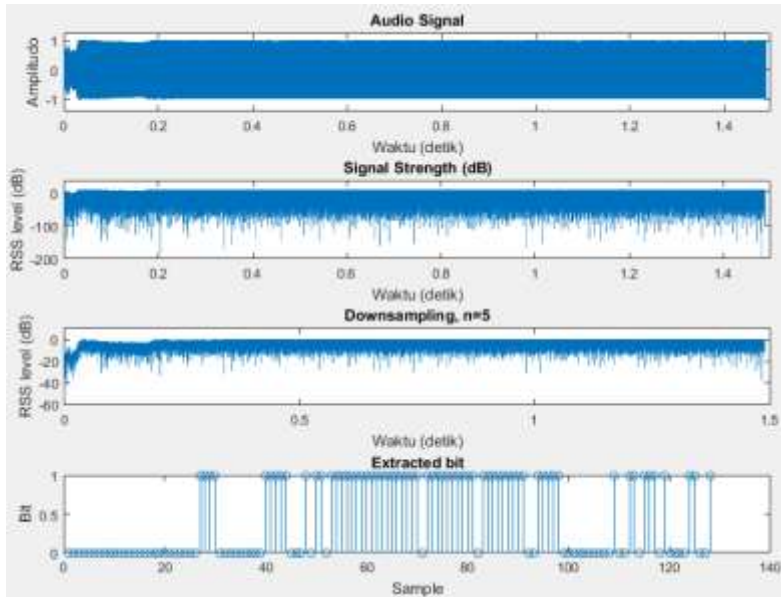


Gambar 4.7 Hasil Pengujian Algoritma Mathur

Bentuk deretan bit yang dihasilkan sekilas hampir sama dengan algoritma sebelumnya, tetapi ada sedikit variasi nilai bit pada awal dan akhir deret bit “1” yang membuat hasil dari kedua algoritma berbeda.

4.3 Hasil Pengujian Lapangan

Pada pengukuran yang telah dilakukan, didapatkan hasil berupa sinyal-sinyal akustik yang nantinya akan diolah ke dalam algoritma yang sudah diuji sebelumnya. Dari dua pasang pengirim dan penerima, didapatkan 7 sinyal dari pasangan pertama (A ke B), dan 9 sinyal dari pasangan kedua (B ke A). Setiap sinyal dari masing-masing pasangan akan dibandingkan dengan kombinasi sebanyak 7×9 atau 63 kombinasi.



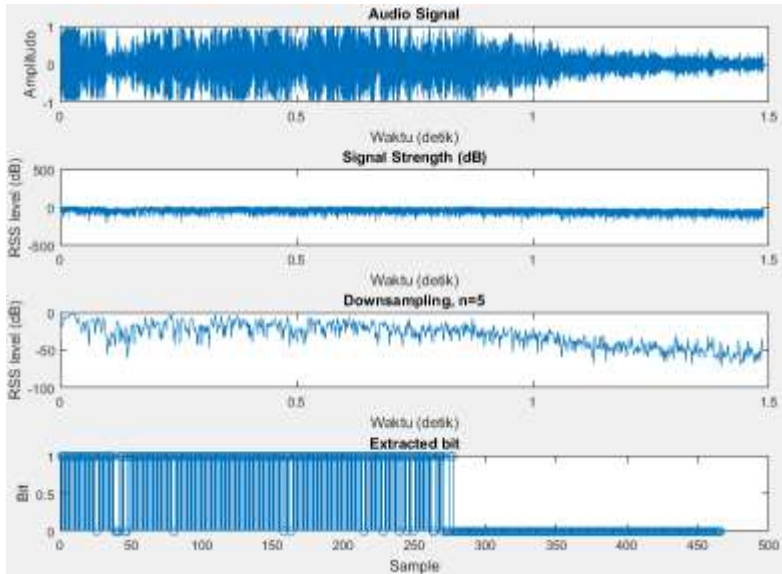
Gambar 4.8 Hasil Ekstraksi Bit menggunakan Algoritma Aono

4.3.1 Hasil Pengolahan Algoritma Aono

Dari semua sinyal hasil pengukuran, masing-masing sinyal akan dimasukkan ke dalam algoritma Aono dengan parameter panjang key 128 bit dan nilai downsampling dari 1 sampai 20. Berikut adalah salah satu hasil dari pengolahan sinyal pengukuran pada algoritma Aono. Jenis sinyal yang dipakai pada contoh pengolahan pada **gambar 4.8** adalah sinyal Tone 5 kHz.

4.3.2 Hasil Pengolahan Algoritma Mathur

Untuk pengolahan sinyal menggunakan algoritma Mathur, parameter yang ditentukan sedikit berbeda. Panjang kunci tidak dapat ditentukan tetapi dapat diubah seiring dengan perubahan nilai downsampling. Untuk mencocokkan bit antar sinyal, perlu dicari terlebih dahulu panjang bit yang sama. Berikut adalah salah satu sinyal hasil pengolahan algoritma Mathur.



Gambar 4.9 Hasil Ekstraksi Bit menggunakan Algoritma Mathur

Hasil yang didapatkan setelah mengolah salah satu sinyal Chirp adalah seperti pada gambar di atas, dengan nilai downsampling 49, panjang bit yang dihasilkan adalah 466 bit.

4.4 Pembahasan

Data yang telah didapatkan dari pengukuran kemudian diolah menggunakan algoritma yang telah ditentukan. Pada subbab ini akan dibahas hasil dari pengolahan sinyal dan juga inti dari tugas akhir ini.

4.4.1 Rekapitulasi Skema Pengujian

Pada tabel berikut adalah jenis-jenis sinyal yang digunakan pada proses pengujian sinyal dan algoritma beserta sinyal yang digunakan pada pengukuran di lapangan.

Tabel 4.1 Rekapitulasi Seluruh Pengujian yang dilakukan

No	Sinyal Input	Pengujian Algoritma	Pengujian Lapangan	Nama Algoritma
1	Tone 1f	V	V	Aono
2	Tone 2f	V	V	
3	Tone 1f var.	V	V	
4	Chirp	V	V	
5	DTMF		V	
6	Noise		V	
1	Tone 1f	V	V	Mathur
2	Tone 2f	V	V	
3	Tone 1f var.	V	V	
4	Chirp	V	V	
5	DTMF		V	
6	Noise		V	

4.4.2 Perhitungan Bit Mismatch Rate

Dari setiap sinyal hasil pengukuran yang berjumlah 7 sinyal dari transmisi pertama dan 9 sinyal dari transmisi kedua, dihasilkan bit secret key. Hasil ini nantinya akan dibandingkan satu per satu dengan pasangan setiap sinyal dari 7 sinyal transmisi pertama dengan setiap sinyal dari 9 sinyal transmisi kedua. Dari total 63 perbandingan akan dipilih kecocokan yang terbaik atau dalam kata lain Bit Mismatch Rate yang terendah.

Ada sedikit perbedaan pada proses pengolahan pada algoritma Aono dan Mathur. Perbedaan terletak pada variasi downsampling dan variasi jumlah bit yang diekstraksi.

Pada algoritma Aono, selain nilai downsampling yang dapat diatur, jumlah bit yang dihasilkan juga dapat ditentukan terlebih dahulu sebelum proses ekstraksi. Hal ini menjadikan variasi pengolahan sinyal menjadi sangat banyak. Dari hasil pengolahan, didapatkan nilai tertentu untuk jumlah bit yang diekstraksi. Pada hasil pengolahan di atas, nilai bit yang digunakan untuk algoritma Aono adalah 128 bit, dengan variasi downsampling bernilai dari 1 sampai 20.

Pada algoritma Mathur, jumlah bit yang dihasilkan tergantung pada panjang dan bentuk sinyal, jadi jumlah bit tidak dapat diatur. Karena pada proses perbandingan jumlah bit harus sama, maka sebelum dibandingkan, panjang bit harus disamakan terlebih dahulu. Proses penyamaan panjang bit dilakukan dengan mencari panjang bit dari

```

bt = zeros(size(bit1));
for j = 1:length(bit1)
    if bit1(j)==bit2(j)
        bt(j)=1;
    end
end
aBMR = 1 - (sum(bt)/length(bt));

```

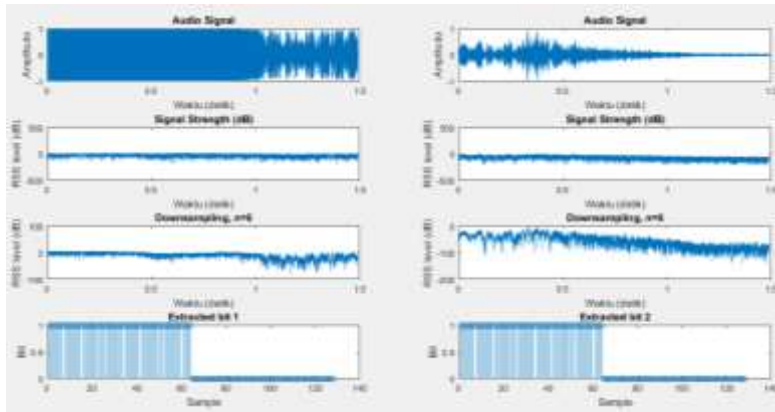
Gambar 4.10 Algoritma perbandingan Bit Secret Key

masing-masing sinyal untuk setiap nilai downsampling. Nilai downsampling yang digunakan adalah 1 sampai 50. Setelah itu, akan dicari nilai panjang bit yang sama dari masing-masing sinyal. Setelah ditemukan panjang bit yang sama, barulah nilai downsampling dari panjang bit tadi digunakan untuk mengolah sinyal agar dapat dibandingkan nilai bitnya. Proses perbandingan secret key didapatkan dengan memasukkan kedua secret key ke dalam algoritma Matlab pada **gambar 4.10**.

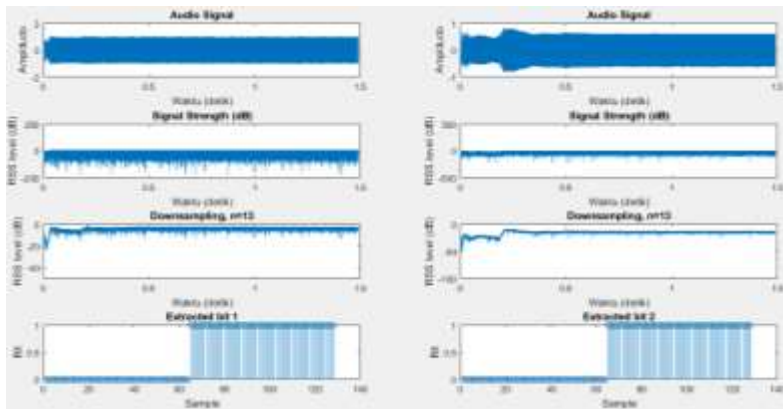
Proses perbandingan secret key hanya dengan membandingkan nilai setiap bit yang berurutan. Jika nilai bit sama, maka akan bernilai 1, jika nilai bit berbeda, maka akan bernilai 0. Deret perbandingan yang baru kemudian dijumlah dan dibagi panjang bit untuk mendapatkan tingkat kecocokannya. Untuk proses perbandingan bit pada Algoritma Mathur sama saja dengan Algoritma Aono.

4.4.2.1 Perbandingan BMR Algoritma Aono

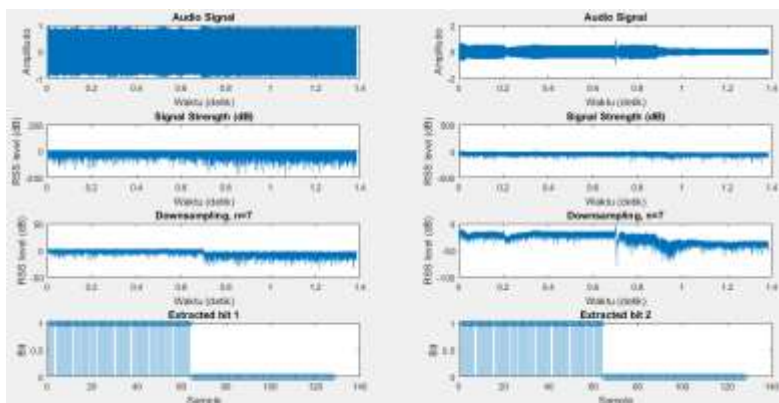
Pada gambar di bawah ini adalah hasil perbandingan setiap jenis sinyal dengan nilai BMR terbaik untuk masing-masing sinyal untuk pengolahan menggunakan Algoritma Aono.



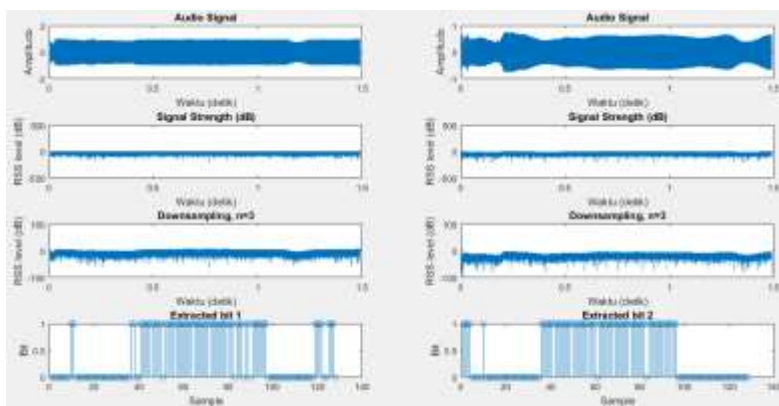
Gambar 4.11 Hasil Ekstraksi Bit algoritma Aono, sinyal Chirp, BMR = 0%



Gambar 4.12 Hasil Ekstraksi Bit algoritma Aono, sinyal Tone 5 kHz, BMR = 0%

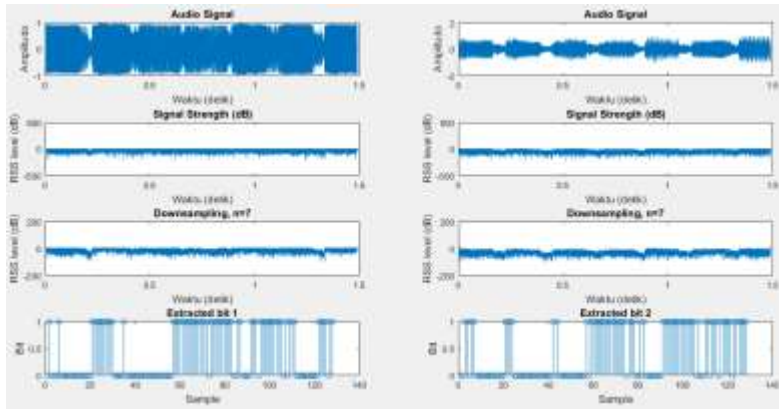


Gambar 4.13 Hasil Ekstraksi Bit algoritma Aono, sinyal Tone 5 kHz & 8 kHz, BMR = 0%

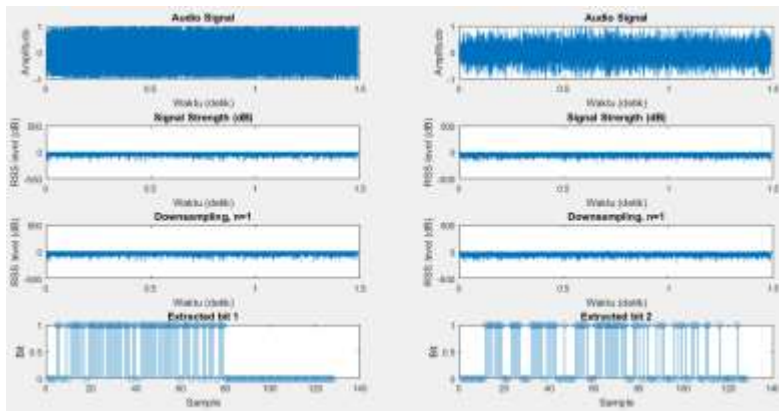


Gambar 4.14 Hasil Ekstraksi Bit algoritma Aono, sinyal Tone 5 kHz bervariasi, BMR = 14,0625%

Dari keempat sinyal di atas, tiga sinyal memiliki nilai BMR sebesar 0%, walaupun nilai ini adalah yang diinginkan, tetapi bentuk deret bit yang ada tidak bervariasi, berbeda dengan sinyal keempat yang deret bitnya lumayan bervariasi, tetapi dengan nilai BMR sebesar 14,0625%.



Gambar 4.15 Hasil Ekstraksi Bit algoritma Aono, sinyal DTMF, BMR
= 26,5625%

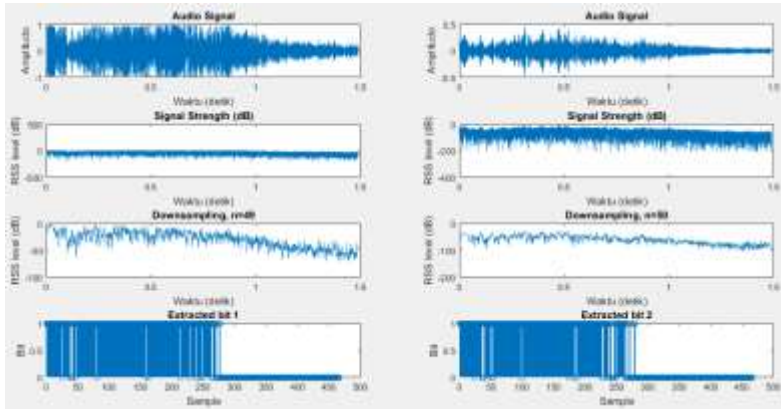


Gambar 4.16 Hasil Ekstraksi Bit algoritma Aono, sinyal White Noise,
BMR = 34,375%

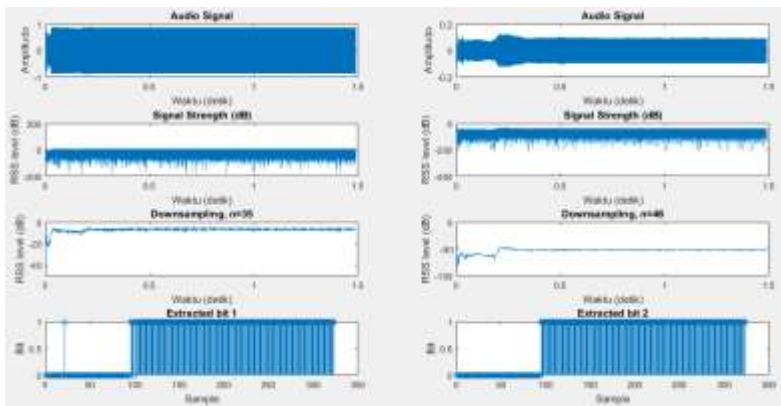
Dari dua sinyal di atas, nilai BMR lebih tinggi dari keempat sinyal sebelumnya, tetapi dengan variasi deret bit yang lebih beragam. Dari enam jenis sinyal yang telah diolah dengan algoritma Aono, tiga sinyal memiliki nilai BMR terendah yaitu sebesar 0%, tetapi dengan bentuk deret bit yang paling sederhana. Sedangkan tiga jenis sinyal lainnya memiliki nilai BMR diatas 10% dengan bentuk deret bit yang bervariasi.

4.4.2.2 Perbandingan BMR Algoritma Mathur

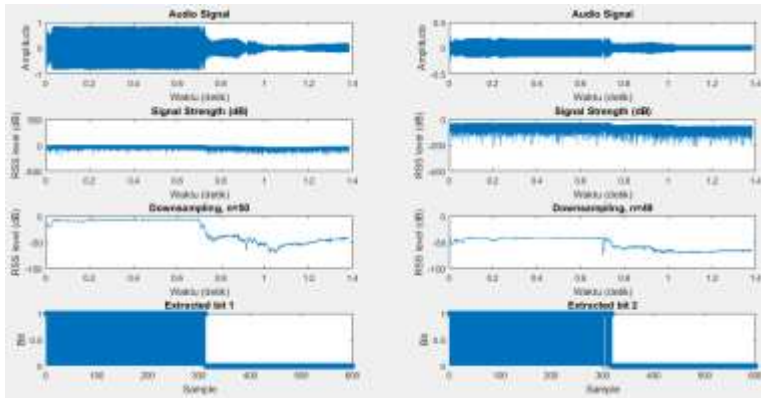
Pada gambar di bawah ini adalah hasil perbandingan setiap jenis sinyal dengan nilai BMR terbaik untuk masing-masing sinyal untuk pengolahan menggunakan Algoritma Mathur.



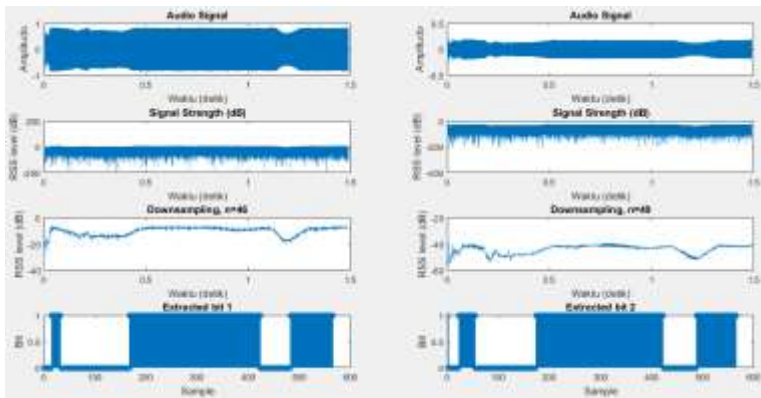
Gambar 4.17 Hasil Ekstraksi Bit algoritma Mathur, sinyal Chirp, bit = 466; BMR = 7,94%



Gambar 4.18 Hasil Ekstraksi Bit algoritma Mathur, sinyal Tone 5 kHz, bit = 323; BMR = 0,929%

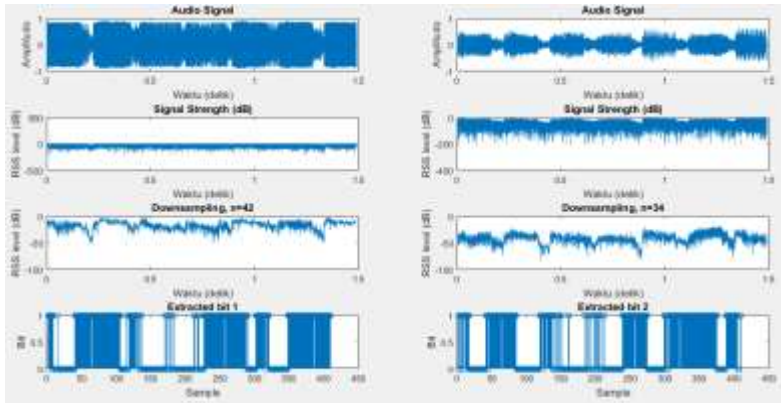


Gambar 4.19 Hasil Ekstraksi Bit algoritma Mathur, sinyal Tone 5 kHz & 8 kHz, bit = 599; BMR = 1,336%

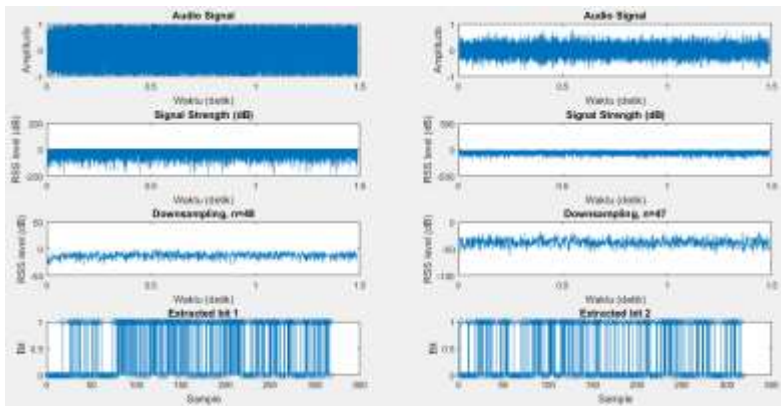


Gambar 4.20 Hasil Ekstraksi Bit algoritma Mathur, sinyal Tone 5 kHz bervariasi, bit = 565; BMR = 7,788%

Dari keempat sinyal di atas, sinyal Tone 5 kHz memiliki nilai BMR paling kecil tetapi dengan bentuk deret bit yang paling sederhana, yaitu deret bit “0” diikuti dengan deret bit “1”. Hal yang sama juga terjadi pada sinyal tone 5 & 8 kHz. Pada sinyal chirp, bentuk deret bit hampir sama dengan sinyal sebelumnya, tetapi terdapat variasi yang lebih banyak, walaupun dengan nilai BMR yang lebih tinggi. Sementara untuk sinyal Tone 5 kHz bervariasi, nilai BMR relatif sama dengan sinyal chirp, dengan deret bit yang terlihat mengikuti bentuk sinyalnya.



Gambar 4.21 Hasil Ekstraksi Bit algoritma Mathur, sinyal DTMF, bit = 409; BMR = 34,963%



Gambar 4.22 Hasil Ekstraksi Bit algoritma Mathur, sinyal White Noise, bit = 317; BMR = 40,063%

Pada dua jenis sinyal terakhir, walaupun bentuk deret bitnya bervariasi, nilai BMR mereka jauh lebih tinggi dari empat sinyal sebelumnya.

Setelah membandingkan semua secret key dari semua jenis sinyal, didapatkan hasil Bit Mismatch Rate terkecil untuk setiap jenis sinyal dan setiap algoritma. Hasilnya dapat dilihat pada **tabel 4.2**.

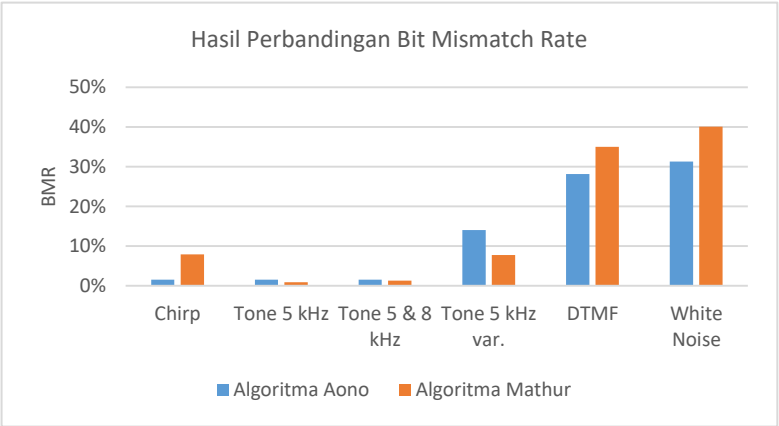
Tabel 4.2 Hasil Perbandingan Bit Mismatch Rate

Jenis Sinyal	Algoritma Aono	Algoritma Mathur
Chirp	1.5625%	7.9399%
Tone 5 kHz	1.5625%	0.9288%
Tone 5 & 8 kHz	1.5625%	1.3356%
Tone 5 kHz var.	14.0625%	7.7876%
DTMF	28.125%	34.9633%
White Noise	31.25%	40.0631%

4.4.3 Perbandingan hasil pengukuran

Dari tabel di atas, hasil Bit Mismatch Rate dapat ditampilkan pada grafik diagram batang seperti berikut ini.

Dari **gambar 4.23** sinyal Tone 5 kHz dan sinyal Tone 5 & 8 kHz memiliki nilai Bit Mismatch Rate terkecil untuk kedua algoritma dibandingkan dengan jenis sinyal lainnya. Sedangkan sinyal DTMF dan sinyal white noise memiliki nilai BMR yang paling besar untuk kedua algoritma jika dibandingkan dengan sinyal lainnya.



Gambar 4.23 Diagram Batang Hasil Perbandingan BMR

BAB 5

PENUTUP

5.1 Kesimpulan

Dari keseluruhan pembahasan tugas akhir tentang sistem komunikasi bawah air yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Secara umum, hasil pengujian algoritma di lab memiliki daya sinyal relatif lebih tinggi dibanding dengan pengukuran di lapangan, dikarenakan redaman yang lebih kecil akibat panjang kanal yang lebih pendek. Perbandingan sinyal diterima dengan referensi 1m pada lab sebesar 0 dB sedangkan pada lapangan sebesar -12dB.
2. Untuk menangkap sinyal yang dikirim lewat sistem komunikasi bawah air diperlukan proses tahapan penangkapan sinyal akustik lewat hydrophone, dikuatkan dengan mixer, dikonversi ke dalam sinyal listrik dan disimpan dalam recorder.
3. Untuk membangkitkan secret key, diperlukan proses pengambilan sinyal dari recorder, dilakukan downsampling, dan kemudian diolah menggunakan algoritma secret key.
4. Pengolahan algoritma secret key dimulai dari memilih durasi sinyal, menghitung nilai RSS, dan kemudian mengekstraksi bit secret key.
5. Dengan sinyal input yang sama, umumnya hasil algoritma Aono lebih baik dari Algoritma Mathur terlihat dari nilai Bit Mismatch Rate terkecil bernilai 1,5625%.
6. Dari semua jenis sinyal yang diujikan, dengan berbagai skema algoritma, dapat disimpulkan bahwa jenis sinyal Tone adalah sinyal yang paling bagus untuk dijadikan sebagai sinyal pembangkit secret key, sedangkan jenis sinyal Noise tidak cocok untuk hal tersebut.

5.2 Saran

Berikut merupakan beberapa saran untuk dilakukan studi dan pengembangan lanjutan untuk penelitian pada tugas akhir ini.

1. Menambahkan jenis sinyal yang lain untuk perbandingan yang lebih banyak.
2. Menambahkan jenis algoritma ekstraksi lain yang dapat mempunyai hasil yang lebih baik dari dua algoritma yang telah diuji.

3. Mengimplementasikan hasil secret key pada suatu sistem enkripsi untuk mengamati performanya.

DAFTAR PUSTAKA

- [1] X. Lurton, An Introduction to Underwater Acoustics. Principles and Applications, New York: Springer, 2010.
- [2] W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed., Pearson, 2017.
- [3] D. W. Pangestu, Wirawan and E. Widjiati, Implementasi Teknik Pengompresian Data dengan Menggunakan Adaptive Huffman Coding untuk Efisiensi Energi Pada Jaringan Sensor Nirkabel Bawah Air (UWSN), Surabaya: JTE-ITS, 2017.
- [4] Y. Luo, L. Pu, Z. Peng and Z. Shi, "RSS-Based Secret Key Generation in Underwater Acoustic Networks: Advantages, Challenges, and Performance Improvements," *IEEE Communications Magazine*, no. February 2016, pp. 32-38, February 2016.

[Halaman ini sengaja dikosongkan]

LAMPIRAN A

LEMBAR PENGESAHAN PROPOSAL

Departemen Teknik Elektro
Fakultas Teknologi Elektro – ITS

EE184801 TUGAS AKHIR – 6 SKS

Nama Mahasiswa	: Tyas Basuki Priambudi
Nomor Pokok	: 07111440000063
Bidang Studi	: Telekomunikasi Multimedia
Tugas Diberikan	: Semester Ganjil 2018/2019
Dosen Pembimbing	: 1. Dr. Ir. Wirawan, DEA. 2. Sri Rahayu, ST., M.Kom.
Judul Tugas Akhir	: Implementasi Pembangkitan Secret Key Berbasis Received Signal Strength (RSS) untuk Sistem Komunikasi Bawah Air (<i>Implementation of RSS Based Secret Key Generation for Underwater Acoustic Communication System</i>)

05 SEP 2018

Uraian Tugas Akhir :

Underwater Acoustic Communication System (UACS) adalah teknologi komunikasi yang banyak digunakan di dunia kemaritiman, namun dewasa ini mulai dieksplorasi untuk kepentingan sistem komunikasi data dengan medium bawah air. Hal ini didorong kenyataan bahwa pemakaian gelombang radio (medium udara) di atas perairan memiliki attenuasi yang lebih besar oleh berbagai faktor peredam, sehingga medium air menjadi alternatif yang cukup potensial. Meskipun demikian, faktor keamanan (*security*) tetap menjadi pertimbangan penting. Ada banyak metode untuk membangun sistem keamanan komunikasi (termasuk di bawah air), diantaranya dengan teknik kriptografi seperti enkripsi (*cipher/text*), autentikasi *server* dan/atau *content*, yang kesemuanya membutuhkan *secret key* yang kuat.

Pada tugas akhir ini, akan diimplementasikan algoritma berbasis *Received Signal Strength (RSS)* untuk pembangkitan *secret key* pada sistem *UACS* sederhana, dengan memilih nilai *RSS* tertentu (khusus) untuk kepentingan sebagai link komunikasi hanya untuk dua pihak itu saja, yang tentu berbeda dari nilai *RSS* link lain. Pembahasan dalam tugas akhir ini meliputi perencanaan link propagasi bawah air (secara teoritis), implementasi *UACS* sederhana di kolam Hidrodinamika ITS, serta pembuatan simulasi pembangkitan *secret key*-nya. Selanjutnya akan dilakukan analisa dan pembahasan untuk kerja sistem *UACS* tersebut, sementara untuk merealisasinya digunakan alat bantu software *Matlab* dan *Asolacity*.

Keywords: *UACS, Security, Autentifikasi, RSS, Secret Key.*

Dosen Pembimbing I,



Dr. Ir. Wirawan, DEA,
NIP. 196311091989031011



Mengesah,
Ketua Program Studi SI

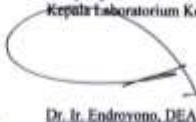
Dr. Ir. Wirawan, ST, M.Eng, Ph.D,
NIP. 197311192000031001

Dosen Pembimbing II,



Sri Rahayu, ST, M.Kom,
NIP. 196802281997022001

Menyetujui,
Kepala Laboratorium Komunikasi Multimedia



Dr. Ir. Endropono, DEA,
NIP. 196504041991021001

[Halaman ini sengaja dikosongkan]

LAMPIRAN B

SKRIP MATLAB

Algoritma Aono :

```
%% Preparation
clear variables
close all
key = 128;
filenamea1 = 'P1capChirp1.mp3';
filenamea2 = 'P1capChirp2.mp3';
filenamea3 = 'P1capChirp3.mp3';
filenamea4 = 'P1capChirp4.mp3';
filenamea5 = 'P2capChirp1.mp3';
filenamea6 = 'P2capChirp2.mp3';
filenamea7 = 'P2capChirp3.mp3';
filenameb1 = 'PM11LChirp.mp3';
filenameb2 = 'PM12LChirp.mp3';
filenameb3 = 'PM13LChirp.mp3';
filenameb4 = 'PM21LChirp.mp3';
filenameb5 = 'PM22LChirp.mp3';
filenameb6 = 'PM23LChirp.mp3';
filenameb7 = 'PM31LChirp.mp3';
filenameb8 = 'PM32LChirp.mp3';
filenameb9 = 'PM33LChirp.mp3';
BMMR=[];

%% Inserting audio file
[ya1,~] = audioread(filenamea1);
[ya2,~] = audioread(filenamea2);
[ya3,~] = audioread(filenamea3);
[ya4,~] = audioread(filenamea4);
[ya5,~] = audioread(filenamea5);
[ya6,~] = audioread(filenamea6);
[ya7,~] = audioread(filenamea7);
[yb1,~] = audioread(filenameb1);
[yb2,~] = audioread(filenameb2);
[yb3,~] = audioread(filenameb3);
[yb4,~] = audioread(filenameb4);
[yb5,~] = audioread(filenameb5);
[yb6,~] = audioread(filenameb6);
[yb7,~] = audioread(filenameb7);
[yb8,~] = audioread(filenameb8);
[yb9,Fs] = audioread(filenameb9);
t=(0:1/Fs:(length(ya1)-1)/Fs);
```

```

ya=[ya1;ya2;ya3;ya4;ya5;ya6;ya7];
yb=[yb1;yb2;yb3;yb4;yb5;yb6;yb7;yb8;yb9];
leny=length(ya1);

%% Signal precutting
for iter=1:length(ya)/leny
    for jter=1:length(yb)/leny
        y1=ya(leny*(iter-1)+1:leny*(iter));
        y2=yb(leny*(jter-1)+1:leny*(jter));
s1 = y1';
fr=1751;
rr=1192;

%% SSR Cutting
s1 = s1(fr:length(s1)-rr);
z1 = ones(size(s1))*1e-9;
sp1 = (s1.^2).*z1;
splog1 = 10*log(sp1);
k1=length(s1);

s2 = y2';
s2 = s2(fr:length(s2)-rr);
z2 = ones(size(s2))*1e-9;
sp2 = (s2.^2).*z2;
splog2 = 10*log(sp2);
bit2=[];
k2=length(s2);

%% RSS Shortening
BMR=[];
for div=1:20
    splog11=[];
    ii=1;
    while ii<(floor(length(splog1)/div)*div)
        splog11a=0;
        for iii=1:div
            splog11a=splog11a+splog1(iii);
            iii=iii+1;
        end
        splog11=[splog11 splog11a/div];
    end
    avg1 = sum(splog11)/length(splog11);

    splog22=[];
    ii=1;
    while ii<(floor(length(splog2)/div)*div)
        splog22a=0;

```

```

        for iii=1:div
            splog22a=splog22a+splog2(ii);
            ii=ii+1;
        end
        splog22=[splog22 splog22a/div];
    end
    avg2 = sum(splog22)/length(splog22);

%% Bit Extraction
    bitr1=zeros(size(splog11));
    RSSmax1=splog11;
    RSSmin1=splog11;
    for i=1:key/2
        [Ma,Ia]=max(RSSmax1);
        bitr1(Ia)=2;
        RSSmax1(Ia)=avg1;
        [Mi,Ii]=min(RSSmin1);
        bitr1(Ii)=1;
        RSSmin1(Ii)=avg1;
    end

    bit1=[];
    for i=1:length(bitr1)
        if bitr1(i) ~= 0
            bit1=[bit1 bitr1(i)];
        end
    end
    bit1=bit1-1;

    bitrp1=bitr1-1;
    bitp1=[];
    for i=1:length(bitrp1)
        if bitrp1(i) ~= -1
            bitp1=[bitp1 bitrp1(i)];
        end
    end

    bitr2=zeros(size(splog22));
    RSSmax2=splog22;
    RSSmin2=splog22;
    for i=1:key/2
        [Mb,Ib]=max(RSSmax2);
        bitr2(Ib)=2;
        RSSmax2(Ib)=avg2;
        [Mj,Ij]=min(RSSmin2);
        bitr2(Ij)=1;
        RSSmin2(Ij)=avg2;
    end

```

```

end

bit2=[];
for i=1:length(bitr2)
    if bitr2(i) ~= 0
        bit2=[bit2 bitr2(i)];
    end
end
bit2=bit2-1;

bitrp2=bitr2-1;
bitp2=[];
for i=1:length(bitrp2)
    if bitrp2(i) ~= -1
        bitp2=[bitp2 bitrp2(i)];
    end
end

%% Bit Mismatch Rate
bt = zeros(size(bitp1));
for j = 1:min(length(bitp1),length(bitp2))
    if bitp1(j)==bitp2(j)
        bt(j)=1;
    end
end
BMR =[BMR (1-(sum(bt)/length(bt)))];

end
BMMR=[BMMR ; BMR];

end

end

```

Algortima Patwari :

```
%% Preparation
clear variables
close all

alfa = 3e-1;
filenamea1 = 'P1capChirp1.mp3';
filenamea2 = 'P1capChirp2.mp3';
filenamea3 = 'P1capChirp3.mp3';
filenamea4 = 'P1capChirp4.mp3';
filenamea5 = 'P2capChirp1.mp3';
filenamea6 = 'P2capChirp2.mp3';
filenamea7 = 'P2capChirp3.mp3';
filenameb1 = 'PM11LChirp.mp3';
filenameb2 = 'PM12LChirp.mp3';
filenameb3 = 'PM13LChirp.mp3';
filenameb4 = 'PM21LChirp.mp3';
filenameb5 = 'PM22LChirp.mp3';
filenameb6 = 'PM23LChirp.mp3';
filenameb7 = 'PM31LChirp.mp3';
filenameb8 = 'PM32LChirp.mp3';
filenameb9 = 'PM33LChirp.mp3';

%% Inserting audio file
[ya1,~] = audioread(filenamea1);
[ya2,~] = audioread(filenamea2);
[ya3,~] = audioread(filenamea3);
[ya4,~] = audioread(filenamea4);
[ya5,~] = audioread(filenamea5);
[ya6,~] = audioread(filenamea6);
[ya7,~] = audioread(filenamea7);
[yb1,~] = audioread(filenameb1);
[yb2,~] = audioread(filenameb2);
[yb3,~] = audioread(filenameb3);
[yb4,~] = audioread(filenameb4);
[yb5,~] = audioread(filenameb5);
[yb6,~] = audioread(filenameb6);
[yb7,~] = audioread(filenameb7);
[yb8,~] = audioread(filenameb8);
[yb9,Fs] = audioread(filenameb9);
t=(0:1/Fs:(length(ya1)-1)/Fs);
ya=[ya1;ya2;ya3;ya4;ya5;ya6;ya7];
yb=[yb1;yb2;yb3;yb4;yb5;yb6;yb7;yb8;yb9];
leny=length(ya1);
```

```

%% Precutting
BMMR=ones(63,10);
co=1;
for iter=1:length(ya)/leny
    for jter=1:length(yb)/leny
        y1=ya(leny*(iter-1)+1:leny*(iter));
        y2=yb(leny*(jter-1)+1:leny*(jter));

s1 = y1';
fr=1751;
rr=1192;

%% Signal cutting
s1 = s1(fr:length(s1)-rr);
z1 = ones(size(s1))*1e-9;
sp1 = (s1.^2).*2+z1;
splog1 = 10*log(sp1);
k1=length(s1);

s2 = y2';
s2 = s2(fr:length(s2)-rr);
z2 = ones(size(s2))*1e-9;
sp2 = (s2.^2).*2+z2;
splog2 = 10*log(sp2);
k2=length(s2);

%% RSS Shortening
lenbit1=[];
lenbit2=[];
for div=1:50
    splog11=[];
    ii=1;
    while ii<(floor(length(splog1)/div)*div)
        splog11a=0;
        for iii=1:div
            splog11a=splog11a+splog1(ii);
            ii=iii+1;
        end
        splog11=[splog11 splog11a/div];
    end
    avg1 = sum(splog11)/length(splog11);
    var1 = sum((splog11-avg1).^2)/length(splog11);
    tau1 = sqrt(var1);
    qup1 = avg1 + alfa*tau1;
    qlw1 = avg1 - alfa*tau1;

```



```

splog22=[];
ii=1;
while ii<(floor(length(splog1)/div)*div)
    splog22a=0;
    for iii=1:div
        splog22a=splog22a+splog2(ii);
        ii=ii+1;
    end
    splog22=[splog22 splog22a/div];
end
avg2 = sum(splog22)/length(splog22);
var2 = sum((splog22-avg2).^2)/length(splog22);
tau2 = sqrt(var2);
qup2 = avg2 + alfa*tau2;
qlw2 = avg2 - alfa*tau2;

%% Bit Extraction and length finding
k11=length(splog11);
i1=1;
bit1=[];
while i1 < k11
    if splog11(i1) > qup1
        if splog11(i1+1) > qup1
            bit1 = [bit1 2];
            i1=i1+2;
        else
            i1=i1+1;
        end
    elseif splog11(i1) < qlw1
        if splog11(i1+1) < qlw1
            bit1 = [bit1 1];
            i1=i1+2;
        else
            i1=i1+1;
        end
    else
        i1=i1+1;
    end
end

k22=length(splog22);
i2=1;
bit2=[];
while i2 < k22
    if splog22(i2) > qup2
        if splog22(i2+1) > qup2
            bit2 = [bit2 2];

```

```

        i2=i2+2;
    else
        i2=i2+1;
    end
elseif splog22(i2) < qlw2
    if splog22(i2+1) < qlw2
        bit2 = [bit2 1];
        i2=i2+2;
    else
        i2=i2+1;
    end
else
    i2=i2+1;
end
end
end

bit1=bit1-1;
lenbit1=[lenbit1 length(bit1)];

bit2=bit2-1;
lenbit2=[lenbit2 length(bit2)];
end
[C,C1,C2]=intersect(lenbit1,lenbit2);

%% Reconstructing bit and Mismatch rate calculation
BMR=[];
for i=1:length(C)
    splog11=[];
    ii=1;
    divd1=C1(i);
    while ii<(floor(length(splog1)/divd1)*divd1)
        splog11a=0;
        for iii=1:divd1
            splog11a=splog11a+splog1(ii);
            ii=ii+1;
        end
        splog11=[splog11 splog11a];
    end
    avg1 = sum(splog11)/length(splog11);
    var1 = sum((splog11-avg1).^2)/length(splog11);
    tau1 = sqrt(var1);
    qup1 = avg1 + alfa*tau1;
    qlw1 = avg1 - alfa*tau1;

    splog22=[];
    ii=1;
    divd2=C2(i);

```

```

while ii<(floor(length(splog1)/divd2)*divd2)
    splog22a=0;
    for iii=1:divd2
        splog22a=splog22a+splog2(ii);
        ii=ii+1;
    end
    splog22=[splog22 splog22a];
end
avg2 = sum(splog22)/length(splog22);
var2 = sum((splog22-avg2).^2)/length(splog22);
tau2 = sqrt(var2);
qup2 = avg2 + alfa*tau2;
qlw2 = avg2 - alfa*tau2;

k11=length(splog11);
i1=1;
bit1=[];
while i1 < k11
    if splog11(i1) > qup1
        if splog11(i1+1) > qup1
            bit1 = [bit1 2];
            i1=i1+2;
        else
            i1=i1+1;
        end
    elseif splog11(i1) < qlw1
        if splog11(i1+1) < qlw1
            bit1 = [bit1 1];
            i1=i1+2;
        else
            i1=i1+1;
        end
    else
        i1=i1+1;
    end
end

k22=length(splog22);
i2=1;
bit2=[];
while i2 < k22
    if splog22(i2) > qup2
        if splog22(i2+1) > qup2
            bit2 = [bit2 2];
            i2=i2+2;
        else
            i2=i2+1;
        end
    end
end

```

```

        end
    elseif splog22(i2) < qlw2
        if splog22(i2+1) < qlw2
            bit2 = [bit2 1];
            i2=i2+2;
        else
            i2=i2+1;
        end
    else
        i2=i2+1;
    end
end

    bt=zeros(size(bit1));
for j = 1:min(length(bit1),length(bit2))
    if bit1(j)==bit2(j)
        bt(j)=1;
    end
end
BMR =[BMR (1-(sum(bt)/length(bt)))];

end
BMMR(co,1:length(BMR))=BMR;
co=co+1;
end
end

```

LAMPIRAN C

TABEL HASIL PENGOLAHAN DATA

A. Algoritma Aono

1. Sinyal Chirp

vs\Div	d1	d2	d6	d12	d13	d17
1v1	0.09375	0.046875	0	0	0.015625	0
1v2	0.125	0.046875	0.015625	0	0.015625	0
1v3	0.09375	0.03125	0.015625	0.03125	0.015625	0
1v4	0.203125	0.09375	0.078125	0.046875	0.078125	0.078125
1v5	0.125	0.078125	0.078125	0.046875	0.0625	0.0625
1v6	0.15625	0.109375	0.03125	0.015625	0.015625	0.015625
1v7	0.140625	0.03125	0.03125	0.015625	0.03125	0.015625
1v8	0.21875	0.09375	0.0625	0.046875	0.046875	0.03125
1v9	0.171875	0.0625	0.015625	0.03125	0.046875	0.03125
2v1	0.09375	0.03125	0	0	0.015625	0
2v2	0.125	0.03125	0.015625	0	0.015625	0
2v3	0.09375	0.03125	0.015625	0.03125	0.015625	0
2v4	0.203125	0.09375	0.078125	0.046875	0.078125	0.078125
2v5	0.140625	0.078125	0.078125	0.046875	0.0625	0.0625
2v6	0.15625	0.109375	0.03125	0.015625	0.015625	0.015625
2v7	0.140625	0.03125	0.03125	0.015625	0.03125	0.015625
2v8	0.21875	0.09375	0.0625	0.046875	0.046875	0.03125
2v9	0.171875	0.0625	0.015625	0.03125	0.046875	0.03125
3v1	0.09375	0.03125	0	0	0.015625	0
3v2	0.125	0.03125	0.015625	0	0.015625	0
3v3	0.09375	0.03125	0.015625	0.03125	0.015625	0
3v4	0.203125	0.078125	0.078125	0.046875	0.078125	0.078125
3v5	0.140625	0.0625	0.078125	0.046875	0.0625	0.0625
3v6	0.15625	0.09375	0.03125	0.015625	0.015625	0.015625
3v7	0.140625	0.03125	0.03125	0.015625	0.03125	0.015625
3v8	0.21875	0.078125	0.0625	0.046875	0.046875	0.03125
3v9	0.171875	0.046875	0.015625	0.03125	0.046875	0.03125
4v1	0.09375	0.0625	0	0	0.015625	0
4v2	0.125	0.0625	0.015625	0	0.015625	0
4v3	0.09375	0.078125	0.015625	0.03125	0.015625	0
4v4	0.203125	0.0625	0.078125	0.046875	0.078125	0.078125

vs\Div	d1	d2	d6	d12	d13	d17
4v5	0.140625	0.0625	0.078125	0.046875	0.0625	0.0625
4v6	0.15625	0.078125	0.03125	0.015625	0.015625	0.015625
4v7	0.140625	0.078125	0.03125	0.015625	0.03125	0.015625
4v8	0.21875	0.078125	0.0625	0.046875	0.046875	0.03125
4v9	0.171875	0.0625	0.015625	0.03125	0.046875	0.03125
5v1	0.15625	0.125	0	0.015625	0.015625	0.015625
5v2	0.125	0.125	0.015625	0.015625	0.015625	0.015625
5v3	0.15625	0.125	0.015625	0.03125	0.015625	0.015625
5v4	0.140625	0.109375	0.078125	0.046875	0.078125	0.078125
5v5	0.15625	0.109375	0.078125	0.046875	0.0625	0.0625
5v6	0.15625	0.125	0.03125	0.015625	0.03125	0.03125
5v7	0.171875	0.125	0.03125	0.015625	0.03125	0.015625
5v8	0.140625	0.125	0.0625	0.046875	0.046875	0.03125
5v9	0.125	0.125	0.015625	0.03125	0.046875	0.03125
6v1	0.125	0.078125	0	0	0.015625	0
6v2	0.109375	0.078125	0.015625	0	0.015625	0
6v3	0.125	0.078125	0.015625	0.03125	0.015625	0
6v4	0.09375	0.109375	0.078125	0.046875	0.078125	0.078125
6v5	0.125	0.09375	0.078125	0.046875	0.0625	0.0625
6v6	0.109375	0.109375	0.03125	0.015625	0.015625	0.015625
6v7	0.125	0.078125	0.03125	0.015625	0.03125	0.015625
6v8	0.09375	0.109375	0.0625	0.046875	0.046875	0.03125
6v9	0.09375	0.078125	0.015625	0.03125	0.046875	0.03125
7v1	0.15625	0.109375	0.03125	0.03125	0.03125	0.03125
7v2	0.140625	0.109375	0.03125	0.03125	0.03125	0.03125
7v3	0.15625	0.109375	0.03125	0.03125	0.03125	0.03125
7v4	0.109375	0.109375	0.078125	0.046875	0.078125	0.078125
7v5	0.171875	0.109375	0.078125	0.046875	0.0625	0.0625
7v6	0.125	0.125	0.046875	0.03125	0.03125	0.03125
7v7	0.171875	0.109375	0.046875	0.03125	0.03125	0.03125
7v8	0.140625	0.09375	0.0625	0.046875	0.046875	0.03125
7v9	0.140625	0.109375	0.03125	0.03125	0.046875	0.03125
Min.	0.09375	0.03125	0	0	0.015625	0

2. Sinyal Tone 5 kHz

vs\Div	d1	d8	d11	d14	d19	d20
1v1	0.53125	0.46875	0.453125	0.421875	0.4375	0.4375
1v2	0.4375	0.390625	0.46875	0.4375	0.4375	0.421875
1v3	0.46875	0.40625	0.453125	0.453125	0.4375	0.4375
1v4	0.4375	0.390625	0.421875	0.4375	0.4375	0.421875
1v5	0.484375	0.40625	0.421875	0.453125	0.421875	0.421875
1v6	0.546875	0.40625	0.421875	0.453125	0.421875	0.421875
1v7	0.40625	0.390625	0.421875	0.515625	0.453125	0.4375
1v8	0.40625	0.4375	0.5	0.484375	0.46875	0.46875
1v9	0.40625	0.375	0.546875	0.484375	0.46875	0.453125
2v1	0.3125	0.484375	0.390625	0.453125	0.46875	0.515625
2v2	0.328125	0.484375	0.421875	0.453125	0.5	0.515625
2v3	0.328125	0.515625	0.453125	0.46875	0.46875	0.515625
2v4	0.328125	0.5	0.484375	0.46875	0.46875	0.53125
2v5	0.34375	0.5	0.46875	0.46875	0.46875	0.53125
2v6	0.3125	0.5	0.46875	0.46875	0.46875	0.53125
2v7	0.328125	0.390625	0.5	0.421875	0.4375	0.484375
2v8	0.328125	0.375	0.390625	0.4375	0.4375	0.484375
2v9	0.328125	0.453125	0.40625	0.453125	0.453125	0.46875
3v1	0.234375	0.46875	0.53125	0.53125	0.5625	0.53125
3v2	0.265625	0.5	0.515625	0.53125	0.546875	0.53125
3v3	0.25	0.46875	0.546875	0.53125	0.5625	0.515625
3v4	0.296875	0.484375	0.515625	0.53125	0.546875	0.53125
3v5	0.28125	0.46875	0.546875	0.546875	0.5625	0.53125
3v6	0.203125	0.46875	0.546875	0.546875	0.5625	0.53125
3v7	0.203125	0.5	0.546875	0.421875	0.53125	0.484375
3v8	0.203125	0.5	0.578125	0.453125	0.5	0.515625
3v9	0.203125	0.484375	0.5	0.46875	0.515625	0.5
4v1	0.5	0.59375	0.53125	0.5	0.46875	0.453125
4v2	0.546875	0.578125	0.53125	0.546875	0.453125	0.4375
4v3	0.578125	0.59375	0.53125	0.5	0.46875	0.4375
4v4	0.578125	0.640625	0.515625	0.53125	0.46875	0.453125
4v5	0.5625	0.640625	0.515625	0.53125	0.46875	0.453125
4v6	0.546875	0.640625	0.515625	0.53125	0.46875	0.453125
4v7	0.453125	0.5625	0.515625	0.46875	0.453125	0.40625

vs\Div	d1	d8	d11	d14	d19	d20
4v8	0.453125	0.625	0.46875	0.46875	0.421875	0.421875
4v9	0.453125	0.546875	0.5	0.4375	0.453125	0.40625
5v1	0.1875	0.25	0.140625	0.09375	0.03125	0.03125
5v2	0.25	0.234375	0.125	0.046875	0.046875	0.015625
5v3	0.234375	0.265625	0.15625	0.125	0.015625	0.078125
5v4	0.296875	0.15625	0.046875	0.03125	0.015625	0.015625
5v5	0.328125	0.140625	0.03125	0.015625	0	0.015625
5v6	0.359375	0.140625	0.03125	0.015625	0	0.015625
5v7	0.21875	0.359375	0.265625	0.390625	0.21875	0.203125
5v8	0.21875	0.34375	0.28125	0.40625	0.234375	0.171875
5v9	0.21875	0.328125	0.3125	0.359375	0.21875	0.15625
6v1	0.21875	0.234375	0.125	0.078125	0.03125	0.09375
6v2	0.28125	0.25	0.125	0.046875	0.046875	0.078125
6v3	0.25	0.25	0.1875	0.109375	0.015625	0.109375
6v4	0.328125	0.171875	0.0625	0.015625	0.015625	0.078125
6v5	0.375	0.15625	0.046875	0.015625	0	0.078125
6v6	0.390625	0.15625	0.046875	0.015625	0	0.078125
6v7	0.171875	0.359375	0.265625	0.390625	0.21875	0.25
6v8	0.171875	0.390625	0.28125	0.390625	0.234375	0.21875
6v9	0.171875	0.390625	0.3125	0.34375	0.21875	0.203125
7v1	0.234375	0.21875	0.140625	0.09375	0.03125	0.046875
7v2	0.25	0.234375	0.125	0.078125	0.046875	0.046875
7v3	0.21875	0.25	0.171875	0.125	0.015625	0.09375
7v4	0.265625	0.125	0.078125	0.0625	0.015625	0.046875
7v5	0.296875	0.109375	0.046875	0.046875	0.015625	0.046875
7v6	0.375	0.109375	0.046875	0.046875	0.015625	0.046875
7v7	0.1875	0.375	0.265625	0.390625	0.21875	0.21875
7v8	0.1875	0.359375	0.3125	0.421875	0.234375	0.1875
7v9	0.1875	0.390625	0.328125	0.359375	0.21875	0.171875
Min.	0.171875	0.109375	0.03125	0.015625	0	0.015625

3. Sinyal Tone 5 kHz & 8 kHz

vsDiv	d1	d5	d6	d9	d14	d17
1v1	0.234375	0.046875	0.0625	0.09375	0.03125	0
1v2	0.328125	0.046875	0.0625	0.109375	0.046875	0
1v3	0.265625	0.046875	0.078125	0.09375	0.03125	0.015625
1v4	0.375	0.28125	0.28125	0.25	0.21875	0.171875
1v5	0.46875	0.203125	0.21875	0.28125	0.171875	0.140625
1v6	0.359375	0.171875	0.1875	0.15625	0.15625	0.171875
1v7	0.234375	0.0625	0.03125	0.015625	0	0
1v8	0.28125	0.078125	0.078125	0.0625	0	0
1v9	0.28125	0.078125	0.046875	0.046875	0	0
2v1	0.28125	0.046875	0.0625	0.09375	0.03125	0
2v2	0.265625	0.046875	0.046875	0.109375	0.046875	0
2v3	0.21875	0.046875	0.078125	0.09375	0.03125	0.015625
2v4	0.375	0.28125	0.296875	0.25	0.21875	0.171875
2v5	0.421875	0.203125	0.21875	0.28125	0.171875	0.140625
2v6	0.28125	0.171875	0.1875	0.15625	0.15625	0.171875
2v7	0.25	0.0625	0.015625	0.015625	0	0
2v8	0.328125	0.078125	0.0625	0.0625	0	0
2v9	0.3125	0.078125	0.03125	0.046875	0	0
3v1	0.203125	0.046875	0.0625	0.09375	0.03125	0
3v2	0.21875	0.046875	0.09375	0.109375	0.046875	0
3v3	0.203125	0.046875	0.078125	0.09375	0.03125	0.015625
3v4	0.40625	0.25	0.28125	0.25	0.21875	0.171875
3v5	0.4375	0.1875	0.203125	0.28125	0.171875	0.140625
3v6	0.3125	0.15625	0.171875	0.15625	0.15625	0.171875
3v7	0.265625	0.078125	0.0625	0.03125	0	0
3v8	0.28125	0.078125	0.09375	0.078125	0	0
3v9	0.28125	0.078125	0.078125	0.0625	0	0
4v1	0.328125	0.046875	0.0625	0.09375	0.03125	0
4v2	0.296875	0.078125	0.078125	0.109375	0.046875	0
4v3	0.3125	0.0625	0.0625	0.09375	0.03125	0.015625
4v4	0.359375	0.234375	0.28125	0.25	0.21875	0.171875
4v5	0.390625	0.171875	0.203125	0.28125	0.171875	0.140625
4v6	0.34375	0.125	0.171875	0.15625	0.15625	0.171875
4v7	0.265625	0.09375	0.0625	0.015625	0	0
4v8	0.234375	0.0625	0.09375	0.0625	0	0
4v9	0.203125	0.0625	0.078125	0.046875	0	0

vs\Div	d1	d5	d6	d9	d14	d17
5v1	0.234375	0.046875	0.0625	0.09375	0.03125	0
5v2	0.234375	0.046875	0.046875	0.109375	0.046875	0
5v3	0.1875	0.046875	0.078125	0.09375	0.03125	0.015625
5v4	0.421875	0.28125	0.296875	0.25	0.21875	0.171875
5v5	0.53125	0.203125	0.21875	0.28125	0.171875	0.140625
5v6	0.375	0.171875	0.1875	0.15625	0.15625	0.171875
5v7	0.234375	0.0625	0.015625	0.015625	0	0
5v8	0.34375	0.078125	0.0625	0.0625	0	0
5v9	0.328125	0.078125	0.03125	0.046875	0	0
6v1	0.203125	0.046875	0.0625	0.09375	0.03125	0
6v2	0.203125	0.046875	0.046875	0.109375	0.046875	0
6v3	0.1875	0.046875	0.078125	0.09375	0.03125	0.015625
6v4	0.46875	0.28125	0.296875	0.25	0.21875	0.171875
6v5	0.5	0.203125	0.21875	0.28125	0.171875	0.140625
6v6	0.375	0.171875	0.1875	0.15625	0.15625	0.171875
6v7	0.234375	0.0625	0.015625	0.015625	0	0
6v8	0.34375	0.078125	0.0625	0.0625	0	0
6v9	0.3125	0.078125	0.03125	0.046875	0	0
7v1	0.25	0.046875	0.0625	0.09375	0.03125	0
7v2	0.3125	0.046875	0.046875	0.109375	0.046875	0
7v3	0.296875	0.046875	0.078125	0.09375	0.03125	0.015625
7v4	0.390625	0.28125	0.296875	0.25	0.21875	0.171875
7v5	0.46875	0.203125	0.21875	0.28125	0.171875	0.140625
7v6	0.28125	0.171875	0.1875	0.15625	0.15625	0.171875
7v7	0.21875	0.0625	0.015625	0.015625	0	0
7v8	0.296875	0.078125	0.0625	0.0625	0	0
7v9	0.296875	0.078125	0.03125	0.046875	0	0
Min.	0.1875	0.046875	0.015625	0.015625	0	0

4. Sinyal Tone 5 kHz bervariasi

vsDiv	d1	d2	d3	d9	d11	d17
1v1	0.5625	0.578125	0.5	0.515625	0.515625	0.5
1v2	0.5625	0.53125	0.609375	0.4375	0.578125	0.5625
1v3	0.5625	0.578125	0.484375	0.5	0.5	0.578125
1v4	0.671875	0.59375	0.5	0.5	0.515625	0.546875
1v5	0.65625	0.59375	0.484375	0.5	0.546875	0.515625
1v6	0.625	0.59375	0.59375	0.5	0.5625	0.53125
1v7	0.59375	0.546875	0.625	0.4375	0.484375	0.5
1v8	0.59375	0.546875	0.578125	0.453125	0.5	0.5
1v9	0.59375	0.5625	0.53125	0.453125	0.484375	0.515625
2v1	0.484375	0.46875	0.390625	0.46875	0.515625	0.4375
2v2	0.453125	0.515625	0.546875	0.4375	0.46875	0.546875
2v3	0.453125	0.484375	0.421875	0.390625	0.453125	0.46875
2v4	0.5	0.546875	0.578125	0.46875	0.453125	0.453125
2v5	0.53125	0.53125	0.5625	0.484375	0.453125	0.4375
2v6	0.484375	0.59375	0.546875	0.515625	0.4375	0.46875
2v7	0.5	0.484375	0.453125	0.46875	0.421875	0.484375
2v8	0.5	0.484375	0.53125	0.5	0.421875	0.484375
2v9	0.5	0.5	0.5	0.421875	0.40625	0.484375
3v1	0.390625	0.40625	0.453125	0.578125	0.5625	0.453125
3v2	0.375	0.359375	0.4375	0.625	0.59375	0.484375
3v3	0.375	0.375	0.421875	0.640625	0.53125	0.5
3v4	0.46875	0.453125	0.4375	0.59375	0.578125	0.5625
3v5	0.484375	0.4375	0.4375	0.59375	0.609375	0.59375
3v6	0.4375	0.4375	0.453125	0.53125	0.59375	0.59375
3v7	0.375	0.375	0.390625	0.5	0.515625	0.609375
3v8	0.375	0.375	0.359375	0.46875	0.578125	0.59375
3v9	0.375	0.390625	0.390625	0.46875	0.5625	0.59375
4v1	0.46875	0.453125	0.46875	0.609375	0.484375	0.484375
4v2	0.515625	0.453125	0.453125	0.546875	0.515625	0.484375
4v3	0.484375	0.5	0.4375	0.5625	0.53125	0.5
4v4	0.484375	0.4375	0.390625	0.53125	0.515625	0.546875
4v5	0.484375	0.4375	0.390625	0.53125	0.515625	0.484375
4v6	0.5	0.5	0.4375	0.53125	0.515625	0.53125
4v7	0.421875	0.453125	0.484375	0.5	0.5	0.546875
4v8	0.421875	0.4375	0.4375	0.515625	0.53125	0.5625
4v9	0.421875	0.453125	0.46875	0.53125	0.5	0.546875

vs\Div	d1	d2	d3	d9	d11	d17
5v1	0.40625	0.5	0.671875	0.390625	0.453125	0.1875
5v2	0.40625	0.515625	0.46875	0.5	0.421875	0.359375
5v3	0.390625	0.53125	0.65625	0.46875	0.328125	0.296875
5v4	0.546875	0.484375	0.328125	0.28125	0.3125	0.234375
5v5	0.546875	0.484375	0.328125	0.265625	0.328125	0.25
5v6	0.484375	0.5625	0.359375	0.28125	0.3125	0.265625
5v7	0.296875	0.59375	0.6875	0.6875	0.625	0.53125
5v8	0.296875	0.59375	0.625	0.6875	0.6875	0.578125
5v9	0.296875	0.59375	0.625	0.703125	0.65625	0.5625
6v1	0.453125	0.53125	0.46875	0.3125	0.34375	0.09375
6v2	0.421875	0.5625	0.453125	0.390625	0.3125	0.265625
6v3	0.40625	0.53125	0.484375	0.34375	0.1875	0.203125
6v4	0.5625	0.625	0.15625	0.3125	0.265625	0.234375
6v5	0.59375	0.65625	0.15625	0.28125	0.296875	0.25
6v6	0.53125	0.703125	0.328125	0.296875	0.296875	0.265625
6v7	0.328125	0.5	0.625	0.78125	0.578125	0.53125
6v8	0.328125	0.515625	0.625	0.796875	0.625	0.578125
6v9	0.328125	0.53125	0.671875	0.796875	0.625	0.5625
7v1	0.296875	0.46875	0.546875	0.359375	0.375	0.109375
7v2	0.3125	0.453125	0.390625	0.4375	0.34375	0.28125
7v3	0.28125	0.421875	0.515625	0.421875	0.21875	0.21875
7v4	0.421875	0.5625	0.140625	0.265625	0.25	0.203125
7v5	0.4375	0.546875	0.15625	0.25	0.28125	0.21875
7v6	0.359375	0.484375	0.3125	0.25	0.265625	0.234375
7v7	0.15625	0.4375	0.640625	0.734375	0.578125	0.5
7v8	0.15625	0.453125	0.546875	0.703125	0.640625	0.546875
7v9	0.15625	0.453125	0.59375	0.75	0.609375	0.53125
Min.	0.15625	0.359375	0.140625	0.25	0.1875	0.09375

5. Sinyal DTMF

vsDiv	d1	d2	d6	d7	d11	d14
1v1	0.59375	0.6875	0.703125	0.453125	0.40625	0.546875
1v2	0.546875	0.625	0.609375	0.515625	0.453125	0.59375
1v3	0.59375	0.71875	0.71875	0.53125	0.453125	0.59375
1v4	0.40625	0.40625	0.328125	0.265625	0.28125	0.421875
1v5	0.40625	0.3125	0.34375	0.296875	0.265625	0.453125
1v6	0.421875	0.40625	0.375	0.296875	0.3125	0.421875
1v7	0.421875	0.4375	0.421875	0.375	0.421875	0.375
1v8	0.4375	0.390625	0.421875	0.40625	0.421875	0.40625
1v9	0.40625	0.359375	0.390625	0.46875	0.4375	0.40625
2v1	0.546875	0.640625	0.734375	0.640625	0.625	0.625
2v2	0.640625	0.578125	0.640625	0.65625	0.640625	0.65625
2v3	0.59375	0.671875	0.671875	0.625	0.65625	0.671875
2v4	0.375	0.40625	0.359375	0.4375	0.40625	0.484375
2v5	0.4375	0.296875	0.359375	0.46875	0.4375	0.484375
2v6	0.421875	0.390625	0.421875	0.453125	0.4375	0.484375
2v7	0.375	0.359375	0.453125	0.4375	0.453125	0.484375
2v8	0.546875	0.421875	0.421875	0.421875	0.390625	0.46875
2v9	0.5625	0.4375	0.421875	0.46875	0.421875	0.515625
3v1	0.671875	0.515625	0.671875	0.4375	0.578125	0.65625
3v2	0.5625	0.46875	0.625	0.484375	0.609375	0.640625
3v3	0.59375	0.515625	0.609375	0.484375	0.671875	0.671875
3v4	0.4375	0.390625	0.34375	0.34375	0.375	0.453125
3v5	0.390625	0.546875	0.40625	0.375	0.46875	0.453125
3v6	0.40625	0.46875	0.4375	0.328125	0.46875	0.421875
3v7	0.4375	0.40625	0.484375	0.484375	0.4375	0.46875
3v8	0.359375	0.453125	0.453125	0.546875	0.40625	0.4375
3v9	0.328125	0.390625	0.46875	0.53125	0.421875	0.5
4v1	0.5625	0.625	0.6875	0.484375	0.5	0.5625
4v2	0.5625	0.578125	0.625	0.5	0.515625	0.640625
4v3	0.609375	0.640625	0.625	0.53125	0.40625	0.609375
4v4	0.4375	0.421875	0.3125	0.34375	0.59375	0.453125
4v5	0.4375	0.390625	0.34375	0.34375	0.5	0.4375
4v6	0.4375	0.484375	0.375	0.3125	0.484375	0.484375
4v7	0.328125	0.46875	0.46875	0.53125	0.5	0.53125
4v8	0.453125	0.453125	0.4375	0.5	0.5	0.53125
4v9	0.453125	0.40625	0.453125	0.53125	0.515625	0.5625

vs\Div	d1	d2	d6	d7	d11	d14
5v1	0.453125	0.453125	0.515625	0.375	0.34375	0.546875
5v2	0.515625	0.484375	0.484375	0.390625	0.359375	0.515625
5v3	0.421875	0.359375	0.515625	0.421875	0.40625	0.53125
5v4	0.453125	0.390625	0.46875	0.46875	0.484375	0.5625
5v5	0.546875	0.5625	0.46875	0.515625	0.5	0.546875
5v6	0.53125	0.46875	0.53125	0.5	0.5	0.59375
5v7	0.609375	0.546875	0.53125	0.390625	0.515625	0.59375
5v8	0.53125	0.5	0.546875	0.515625	0.53125	0.609375
5v9	0.515625	0.484375	0.515625	0.5	0.515625	0.53125
6v1	0.4375	0.359375	0.546875	0.328125	0.34375	0.5625
6v2	0.4375	0.390625	0.5	0.28125	0.375	0.53125
6v3	0.28125	0.3125	0.5	0.375	0.453125	0.546875
6v4	0.4375	0.53125	0.46875	0.53125	0.4375	0.703125
6v5	0.5	0.65625	0.515625	0.53125	0.515625	0.703125
6v6	0.484375	0.5625	0.609375	0.5625	0.5	0.71875
6v7	0.609375	0.578125	0.609375	0.421875	0.46875	0.65625
6v8	0.453125	0.578125	0.65625	0.546875	0.546875	0.65625
6v9	0.453125	0.59375	0.640625	0.546875	0.5	0.640625
7v1	0.515625	0.5	0.453125	0.3125	0.28125	0.5
7v2	0.515625	0.53125	0.40625	0.265625	0.3125	0.53125
7v3	0.46875	0.421875	0.421875	0.296875	0.421875	0.53125
7v4	0.359375	0.421875	0.5625	0.578125	0.484375	0.625
7v5	0.4375	0.53125	0.546875	0.5625	0.5625	0.609375
7v6	0.421875	0.4375	0.546875	0.5625	0.546875	0.65625
7v7	0.5	0.515625	0.578125	0.4375	0.53125	0.625
7v8	0.453125	0.46875	0.546875	0.546875	0.578125	0.640625
7v9	0.453125	0.46875	0.578125	0.53125	0.546875	0.59375
Min.	0.28125	0.296875	0.3125	0.265625	0.265625	0.375

6. Sinyal Noise

vsDiv	d1	d2	d3	d5	d13	d14
1v1	0.421875	0.53125	0.515625	0.53125	0.515625	0.5
1v2	0.390625	0.515625	0.546875	0.4375	0.53125	0.453125
1v3	0.484375	0.46875	0.484375	0.5	0.4375	0.453125
1v4	0.390625	0.484375	0.484375	0.578125	0.453125	0.40625
1v5	0.46875	0.484375	0.484375	0.46875	0.4375	0.515625
1v6	0.453125	0.5	0.46875	0.484375	0.53125	0.453125
1v7	0.453125	0.515625	0.484375	0.5	0.5	0.453125
1v8	0.421875	0.484375	0.390625	0.5	0.5	0.515625
1v9	0.515625	0.515625	0.515625	0.421875	0.421875	0.484375
2v1	0.40625	0.515625	0.53125	0.5	0.46875	0.46875
2v2	0.375	0.421875	0.484375	0.46875	0.40625	0.515625
2v3	0.453125	0.484375	0.421875	0.421875	0.53125	0.5
2v4	0.453125	0.46875	0.53125	0.53125	0.46875	0.46875
2v5	0.453125	0.46875	0.515625	0.46875	0.46875	0.515625
2v6	0.484375	0.53125	0.515625	0.5	0.515625	0.5625
2v7	0.5	0.53125	0.46875	0.5	0.421875	0.453125
2v8	0.375	0.46875	0.546875	0.53125	0.46875	0.453125
2v9	0.515625	0.484375	0.515625	0.5	0.453125	0.5625
3v1	0.40625	0.453125	0.515625	0.40625	0.578125	0.484375
3v2	0.421875	0.484375	0.4375	0.4375	0.515625	0.484375
3v3	0.4375	0.484375	0.46875	0.515625	0.46875	0.46875
3v4	0.46875	0.453125	0.484375	0.53125	0.515625	0.5625
3v5	0.484375	0.53125	0.484375	0.5625	0.515625	0.515625
3v6	0.46875	0.546875	0.546875	0.453125	0.546875	0.484375
3v7	0.484375	0.5	0.5	0.40625	0.53125	0.53125
3v8	0.40625	0.546875	0.4375	0.46875	0.546875	0.515625
3v9	0.515625	0.546875	0.453125	0.46875	0.546875	0.578125
4v1	0.390625	0.453125	0.40625	0.515625	0.578125	0.359375
4v2	0.421875	0.46875	0.546875	0.515625	0.46875	0.390625
4v3	0.40625	0.4375	0.453125	0.46875	0.46875	0.375
4v4	0.453125	0.46875	0.484375	0.53125	0.453125	0.46875
4v5	0.453125	0.453125	0.5	0.46875	0.453125	0.484375
4v6	0.5	0.5	0.484375	0.453125	0.546875	0.515625
4v7	0.53125	0.484375	0.46875	0.4375	0.4375	0.546875
4v8	0.421875	0.5	0.5	0.484375	0.5	0.59375
4v9	0.53125	0.453125	0.515625	0.453125	0.484375	0.546875

vs\Div	d1	d2	d3	d5	d13	d14
5v1	0.421875	0.453125	0.46875	0.453125	0.484375	0.375
5v2	0.390625	0.515625	0.5	0.4375	0.453125	0.375
5v3	0.453125	0.515625	0.5	0.5	0.421875	0.421875
5v4	0.40625	0.484375	0.46875	0.515625	0.421875	0.484375
5v5	0.484375	0.453125	0.390625	0.578125	0.59375	0.515625
5v6	0.5	0.484375	0.4375	0.484375	0.53125	0.53125
5v7	0.40625	0.46875	0.53125	0.453125	0.515625	0.5
5v8	0.375	0.46875	0.4375	0.421875	0.515625	0.53125
5v9	0.4375	0.5	0.453125	0.609375	0.53125	0.53125
6v1	0.390625	0.5	0.4375	0.390625	0.421875	0.515625
6v2	0.390625	0.421875	0.421875	0.5	0.421875	0.5
6v3	0.375	0.53125	0.3125	0.46875	0.484375	0.515625
6v4	0.421875	0.484375	0.4375	0.34375	0.5	0.375
6v5	0.46875	0.4375	0.421875	0.421875	0.46875	0.46875
6v6	0.484375	0.53125	0.484375	0.5	0.46875	0.5
6v7	0.453125	0.5	0.4375	0.53125	0.5	0.484375
6v8	0.375	0.484375	0.5	0.5	0.5	0.515625
6v9	0.4375	0.5	0.4375	0.46875	0.5625	0.546875
7v1	0.34375	0.46875	0.40625	0.453125	0.453125	0.46875
7v2	0.359375	0.421875	0.453125	0.453125	0.46875	0.40625
7v3	0.390625	0.515625	0.40625	0.421875	0.5	0.5
7v4	0.40625	0.4375	0.5	0.484375	0.4375	0.421875
7v5	0.5	0.421875	0.5	0.5	0.421875	0.453125
7v6	0.5	0.4375	0.484375	0.5	0.46875	0.5
7v7	0.484375	0.46875	0.5625	0.453125	0.5	0.515625
7v8	0.40625	0.515625	0.59375	0.5	0.484375	0.46875
7v9	0.4375	0.5	0.484375	0.46875	0.5625	0.5
Min.	0.34375	0.421875	0.3125	0.34375	0.40625	0.359375

B. Algoritma Patwari

1. Sinyal Chirp

vs\no.	n1	n2	n3	n4
1v1	0.155597723	0.158769742	1	1
1v2	1	1	1	1
1v3	0.140039448	0.142512077	1	1
1v4	1	1	1	1
1v5	0.130952381	1	1	1
1v6	1	1	1	1
1v7	0.128700129	1	1	1
1v8	1	1	1	1
1v9	0.141353383	0.135135135	1	1
2v1	0.136622391	1	1	1
2v2	1	1	1	1
2v3	0.136094675	0.136678201	1	1
2v4	0.132921175	1	1	1
2v5	0.128284389	1	1	1
2v6	0.13229572	0.142687277	1	1
2v7	0.124100719	1	1	1
2v8	0.117723157	1	1	1
2v9	0.12763596	1	1	1
3v1	0.160147601	1	1	1
v32	0.148496241	0.148550725	1	1
3v3	0.148550725	0.140138408	0.149206349	0.150147203
3v4	0.12849162	0.133928571	0.1443769	1
3v5	0.142857143	1	1	1
3v6	0.130353818	0.139968896	1	1
3v7	0.120350109	0.123214286	0.125613346	1
3v8	0.135399674	0.133738602	0.128275862	1
3v9	1	1	1	1
4v1	0.142414861	0.157894737	1	1
4v2	0.144927536	0.146010187	0.145325203	1
4v3	0.144927536	1	1	1
4v4	0.136546185	0.143626571	1	1
4v5	1	1	1	1
4v6	0.154761905	0.144970414	1	1
4v7	0.128421053	1	1	1
4v8	0.134538153	0.130904184	0.117886179	1

vs\no.	n1	n2	n3	n4
4v9	0.123931624	0.147217235	1	1
5v1	0.110679612	0.113475177	1	1
5v2	0.1	1	1	1
5v3	0.098619329	0.103846154	0.116883117	1
5v4	0.130353818	0.136231884	1	1
5v5	0.126959248	0.130702836	1	1
5v6	0.124767225	0.127527216	0.131793478	1
5v7	0.11525974	0.091377091	0.163363363	1
5v8	0.101659751	0.117241379	1	1
5v9	0.110682111	0.150379805	1	1
6v1	0.102466793	1	1	1
6v2	0.112318841	0.108882521	1	1
6v3	0.112318841	0.112432432	1	1
6v4	0.109126984	0.116071429	0.139931741	0.130372493
6v5	1	1	1	1
6v6	0.119047619	1	1	1
6v7	0.101785714	0.107449857	0.099863201	1
6v8	0.10805501	0.104056437	0.091819699	1
6v9	0.096828047	0.10075188	0.104854369	1
7v1	0.096491228	1	1	1
7v2	1	1	1	1
7v3	0.096646943	0.100371747	0.104022191	0.106014271
7v4	0.110714286	0.124802528	1	1
7v5	0.1133829	1	1	1
7v6	0.118483412	1	1	1
7v7	0.085714286	0.105882353	1	1
7v8	0.102292769	0.091627172	1	1
7v9	0.079399142	1	1	1
Min.	0.079399142			

2. Sinyal Tone 5 kHz

vs/no.	n1	n2	n3	n4
1v1	0.439130435	0.394648829	1	1
1v2	0.421259843	0.41958042	0.424749164	0.387019231
1v3	0.473913043	0.340757238	1	1
1v4	1	1	1	1
1v5	0.401215805	0.408719346	0.390625	0.47020934
1v6	0.388429752	0.395238095	1	1
1v7	0.377952756	0.368794326	0.391752577	0.442446043
1v8	0.408695652	0.412	0.367483296	0.37593985
1v9	0.357692308	0.369565217	0.371900826	1
2v1	0.338842975	1	1	1
2v2	0.41958042	1	1	1
2v3	0.440613027	0.461538462	1	1
2v4	0.395721925	0.39095315	1	1
2v5	0.410778443	1	1	1
2v6	0.373809524	1	1	1
2v7	0.423791822	1	1	1
2v8	0.394255875	1	1	1
2v9	0.463768116	0.513416816	1	1
3v1	0.512396694	0.481239804	1	1
v32	0.465568862	1	1	1
3v3	0.490990991	1	1	1
3v4	0.363636364	1	1	1
3v5	0.477707006	0.422651934	0.417069243	1
3v6	1	1	1	1
3v7	0.530499076	1	1	1
3v8	0.469465649	0.441064639	0.455782313	0.43575419
3v9	0.527896996	0.5	0.485507246	0.476047904
4v1	0.462616822	0.445378151	1	1
4v2	0.411290323	0.565737052	0.343661972	1
4v3	1	1	1	1
4v4	0.552855407	1	1	1
4v5	0.433701657	0.483685221	0.47873633	1
4v6	1	1	1	1
4v7	0.473029046	0.494023904	0.47440273	0.424501425
4v8	0.422053232	0.441176471	1	1
4v9	0.493775934	0.452883263	1	1

vs\no.	n1	n2	n3	n4
5v1	0.090909091	0.226190476	1	1
5v2	0.102941176	0.239837398	0.150645624	1
5v3	0.118644068	0.119402985	0.101587302	0.26541555
5v4	0.053078556	1	1	1
5v5	0.055201699	0.092631579	1	1
5v6	0.056	0.28358209	1	1
5v7	0.149372108	1	1	1
5v8	0.187319885	1	1	1
5v9	0.112318841	0.152815013	0.092039801	1
6v1	0.068421053	0.287735849	0.147679325	1
6v2	0.074418605	0.066929134	1	1
6v3	0.113513514	0.175463623	1	1
6v4	0.009287926	0.114228457	0.44835966	1
6v5	0.181052632	0.182260024	1	1
6v6	1	1	1	1
6v7	0.228346457	1	1	1
6v8	0.132352941	0.118705036	1	1
6v9	0.21574344	0.146371464	1	1
7v1	0.09569378	0.098130841	1	1
7v2	0.123348018	0.200787402	1	1
7v3	0.151351351	0.20242915	1	1
7v4	0.050104384	1	1	1
7v5	1	1	1	1
7v6	1	1	1	1
7v7	0.076923077	0.094488189	0.077777778	0.147727273
7v8	1	1	1	1
7v9	1	1	1	1
Min.	0.009287926			

3. Sinyal Tone 5 kHz & 8 kHz

vs/no.	n1	n2	n3	n4
1v1	1	1	1	1
1v2	0.027027027	0.025550661	1	1
1v3	0.014647137	0.021664766	1	1
1v4	0.143164693	0.134013605	1	1
1v5	0.150492264	1	1	1
1v6	0.135922233	0.155074116	1	1
1v7	0.058343057	0.030612245	1	1
1v8	0.064189189	0.051519155	0.058343057	0.053846154
1v9	1	1	1	1
2v1	0.022959184	1	1	1
2v2	0.025222552	1	1	1
2v3	1	1	1	1
2v4	0.163892446	1	1	1
2v5	0.154205607	1	1	1
2v6	0.143431635	1	1	1
2v7	1	1	1	1
2v8	0.055007052	1	1	1
2v9	0.049559471	1	1	1
3v1	1	1	1	1
v32	0.022764228	0.021341463	0.022099448	1
3v3	1	1	1	1
3v4	1	1	1	1
3v5	0.141625616	0.1349134	1	1
3v6	0.137628111	0.14147018	0.105914718	1
3v7	1	1	1	1
3v8	0.061904762	0.051546392	0.052763819	1
3v9	0.054858934	1	1	1
4v1	0.023255814	0.021978022	0.043346337	1
4v2	0.026696329	1	1	1
4v3	0.018158236	0.019038076	1	1
4v4	0.167692308	1	1	1
4v5	1	1	1	1
4v6	0.134615385	1	1	1
4v7	1	1	1	1
4v8	1	1	1	1
4v9	0.052307692	1	1	1

vs\no.	n1	n2	n3	n4
5v1	0.021374046	1	1	1
5v2	0.023217247	0.025559105	0.024960998	1
5v3	0.013740458	0.017167382	0.015564202	0.016460905
5v4	0.136898396	1	1	1
5v5	0.158798283	0.14408867	1	1
5v6	0.145936982	0.137285491	0.134699854	1
5v7	0.051841746	1	1	1
5v8	0.054570259	1	1	1
5v9	0.053964758	1	1	1
6v1	1	1	1	1
6v2	1	1	1	1
6v3	0.019859813	0.021235521	1	1
6v4	0.171288744	0.165625	0.160123967	1
6v5	0.150046598	1	1	1
6v6	0.133498146	1	1	1
6v7	0.032328528	1	1	1
6v8	0.055464927	1	1	1
6v9	0.057104914	0.053964758	1	1
7v1	1	1	1	1
7v2	1	1	1	1
7v3	0.013355593	0.015649452	0.014662757	0.015068493
7v4	0.173622705	0.162190083	1	1
7v5	1	1	1	1
7v6	0.144155844	1	1	1
7v7	0.055718475	0.05316092	1	1
7v8	0.057471264	1	1	1
7v9	0.05509182	0.054098361	0.059036145	1
Min.	0.013355593			

4. Sinyal Tone 5 kHz bervariasi

vs/no.	n1	n2	n3	n4
1v1	0.397748593	0.4	1	1
1v2	0.459701493	1	1	1
1v3	1	1	1	1
1v4	1	1	1	1
1v5	0.519685039	1	1	1
1v6	0.5	0.519402985	1	1
1v7	0.442647059	0.471561531	1	1
1v8	1	1	1	1
1v9	0.438655462	0.451913133	1	1
2v1	0.330721003	0.357675112	1	1
2v2	1	1	1	1
2v3	0.350223547	0.382992749	1	1
2v4	0.452459016	0.457388585	1	1
2v5	0.460655738	0.502362205	0.443037975	1
2v6	1	1	1	1
2v7	0.292134831	0.330868762	0.302795031	1
2v8	1	1	1	1
2v9	0.385457389	1	1	1
3v1	0.405357143	0.428571429	1	1
v32	0.4125	0.412474849	1	1
3v3	0.423550088	0.41750503	1	1
3v4	1	1	1	1
3v5	0.471042471	0.5	1	1
3v6	0.458536585	1	1	1
3v7	1	1	1	1
3v8	0.390738061	1	1	1
3v9	0.401785714	0.430252101	0.380607815	1
4v1	0.355371901	1	1	1
4v2	0.361983471	0.416913947	0.459343795	1
4v3	0.36970684	1	1	1
4v4	0.451239669	1	1	1
4v5	0.471698113	1	1	1
4v6	0.461254613	1	1	1
4v7	0.365079365	0.373382625	0.39049236	1
4v8	0.371549894	1	1	1
4v9	1	1	1	1

vs\no.	n1	n2	n3	n4
5v1	0.079734219	0.1359447	1	1
5v2	0.077876106	0.100342075	0.181818182	1
5v3	0.122123894	1	1	1
5v4	0.417699115	0.413675214	0.404761905	0.397280967
5v5	1	1	1	1
5v6	0.431734317	1	1	1
5v7	0.136363636	0.119565217	1	1
5v8	0.149171271	0.162436548	0.253261704	1
5v9	1	1	1	1
6v1	0.127810651	1	1	1
6v2	1	1	1	1
6v3	0.122383253	1	1	1
6v4	0.404509284	0.419653179	0.382084095	1
6v5	1	1	1	1
6v6	0.4256	1	1	1
6v7	0.138408304	0.101626016	1	1
6v8	0.158579882	1	1	1
6v9	0.183783784	0.171192444	1	1
7v1	0.107394366	0.10887574	0.119375574	1
7v2	1	1	1	1
7v3	1	1	1	1
7v4	1	1	1	1
7v5	1	1	1	1
7v6	1	1	1	1
7v7	0.120147874	0.12	0.117540687	0.086821705
7v8	0.133671743	0.144378698	0.112947658	1
7v9	0.174774775	1	1	1
Min.	0.077876106			

5. Sinyal DTMF

vs\no.	n1	n2	n3	n4
1v1	0.458823529	0.410404624	0.448179272	0.376963351
1v2	0.441791045	0.428176796	0.5	0.394101877
1v3	0.486404834	0.476190476	0.487980769	0.531194296
1v4	0.43697479	0.44198895	0.418230563	0.447643979
1v5	0.420765027	0.421465969	0.440074906	1
1v6	1	1	1	1
1v7	0.450819672	1	1	1
1v8	0.501510574	0.488950276	1	1
1v9	0.473988439	0.486187845	0.477211796	0.46997389
2v1	0.484848485	0.504201681	1	1
2v2	0.530831099	0.519704433	0.513636364	1
2v3	0.41091954	0.481375358	0.515406162	1
2v4	0.431372549	0.394101877	0.459584296	1
2v5	0.385224274	1	1	1
2v6	0.41260745	0.421446384	1	1
2v7	0.408775982	1	1	1
2v8	0.474474474	0.501432665	0.458300238	1
2v9	0.44772118	0.462140992	0.413636364	1
3v1	1	1	1	1
v32	0.408571429	0.541966427	1	1
3v3	0.475903614	0.41091954	0.522781775	1
3v4	0.449008499	1	1	1
3v5	0.420634921	0.415	0.446650124	0.384976526
3v6	0.418981481	0.424187726	1	1
3v7	0.497142857	0.468144044	0.4075	0.444664032
3v8	0.388429752	1	1	1
3v9	1	1	1	1
4v1	0.389534884	0.489637306	0.513002364	0.512717537
4v2	0.528497409	0.498817967	1	1
4v3	0.472049689	0.472891566	0.454545455	1
4v4	1	1	1	1
4v5	0.430079156	0.433931485	1	1
4v6	0.411471322	0.418981481	0.44045677	0.435612083
4v7	0.431767338	0.440298507	1	1
4v8	0.39675174	0.446124764	1	1
4v9	0.459530026	0.445520581	1	1

vs\no.	n1	n2	n3	n4
5v1	0.350241546	0.389550871	1	1
5v2	0.439678284	0.420454545	0.466184822	1
5v3	0.380681818	0.38588451	1	1
5v4	0.528150134	0.492822967	0.470862471	0.555555556
5v5	0.509234828	0.5075	0.504784689	0.521384929
5v6	0.546492659	0.543720191	1	1
5v7	0.53	0.523923445	1	1
5v8	0.520661157	0.533980583	0.511111111	0.554404145
5v9	0.588732394	0.576407507	0.525	1
6v1	0.368983957	1	1	1
6v2	0.431428571	0.45308311	0.36453202	1
6v3	1	1	1	1
6v4	0.525469169	0.521052632	0.556675063	0.527802295
6v5	0.521091811	0.525386313	0.558044807	0.505813953
6v6	0.542780749	0.513157895	0.572319202	1
6v7	0.594285714	0.526960784	0.550321199	1
6v8	1	1	1	1
6v9	0.568364611	0.537914692	0.517006803	0.55906822
7v1	0.349633252	0.368794326	0.461946903	1
7v2	0.394101877	0.359338061	1	1
7v3	0.415472779	0.395275591	1	1
7v4	0.482573727	0.513853904	0.531111111	1
7v5	1	1	1	1
7v6	0.535816619	1	1	1
7v7	1	1	1	1
7v8	0.573065903	0.511627907	0.504444444	0.573127229
7v9	0.54691689	0.562347188	0.558858502	0.566886623
Min.	0.349633252			

6. Sinyal Noise

vs\no.	n1	n2	n3	n4
1v1	0.507389163	0.471968709	1	1
1v2	0.531976744	0.520942408	0.537037037	1
1v3	0.512328767	0.459049545	1	1
1v4	0.460273973	0.462962963	0.50965251	1
1v5	0.5	0.488069414	0.471494607	0.474576271
1v6	0.5	0.5	1	1
1v7	0.494594595	0.497340426	0.508196721	1
1v8	0.450402145	0.497991968	1	1
1v9	0.488745981	0.505813953	1	1
2v1	0.508090615	0.528700906	0.507399577	1
2v2	0.465608466	0.5	0.492574257	0.489278752
2v3	0.481578947	1	1	1
2v4	0.46366782	0.488178025	0.49249531	1
2v5	0.474320242	0.458115183	0.471869328	0.494363929
2v6	0.48630137	0.482926829	1	1
2v7	0.509933775	0.540540541	0.517857143	0.486257928
2v8	1	1	1	1
2v9	0.494023904	0.525663717	1	1
3v1	0.454751131	0.47148289	0.478688525	1
v32	0.49127907	1	1	1
3v3	0.494305239	1	1	1
3v4	0.498269896	1	1	1
3v5	0.544615385	1	1	1
3v6	0.484375	0.504615385	0.476744186	1
3v7	0.483425414	0.460076046	0.467463479	1
3v8	1	1	1	1
3v9	0.450160772	0.503125	0.459302326	0.506925208
4v1	0.459276018	0.522	1	1
4v2	0.47382199	1	1	1
4v3	0.426751592	0.472222222	0.476082005	0.530956848
4v4	0.512738854	0.483009709	1	1
4v5	0.488448845	0.510471204	1	1
4v6	0.5	0.461538462	1	1
4v7	0.438943894	0.458598726	1	1
4v8	0.465277778	0.442367601	0.492359932	1
4v9	0.481481481	0.476555024	1	1

vs\no.	n1	n2	n3	n4
5v1	0.422712934	1	1	1
5v2	0.432352941	0.424418605	0.475452196	1
5v3	0.473477407	1	1	1
5v4	0.400630915	1	1	1
5v5	1	1	1	1
5v6	0.415697674	0.493123772	0.427549195	1
5v7	1	1	1	1
5v8	0.492505353	1	1	1
5v9	0.504373178	0.456395349	1	1
6v1	0.429022082	1	1	1
6v2	0.423529412	0.459530026	0.440545809	0.476722533
6v3	1	1	1	1
6v4	0.495268139	0.463592233	0.485714286	1
6v5	0.523809524	1	1	1
6v6	0.482876712	0.503579952	0.508305648	1
6v7	1	1	1	1
6v8	0.444089457	1	1	1
6v9	0.442295293	1	1	1
7v1	0.415384615	0.425855513	0.458074534	1
7v2	1	1	1	1
7v3	0.447530864	1	1	1
7v4	0.488178025	0.471398305	1	1
7v5	0.488448845	0.449275362	0.469810236	1
7v6	1	1	1	1
7v7	0.486754967	0.419141914	0.468822171	0.467680608
7v8	0.488817891	1	1	1
7v9	0.494923858	0.516483516	0.452205882	1
Min.	0.400630915			

BIODATA PENULIS



Tyas Basuki Priambudi. Lahir 1 April 1997 di Surabaya. Anak tunggal yang hidup di tengah keluarga kecil sederhana. Pria dengan kegemaran membaca dan menonton film ini menempuh jenjang pendidikan S1 Teknik Elektro ITS setelah lulus dari SMAN 1 Sidoarjo pada tahun 2014, hingga pada akhir tahun 2018 menyelesaikan studinya dengan mengikuti seminar dan sidang tugas akhir sebagai persyaratan kelulusan dan mendapatkan gelar sarjana.